



TUGAS AKHIR - KI141502

**RANCANG BANGUN SIMULASI UJIAN SURAT
IZIN MENGEMUDI (SIM) MOBIL
MENGUNAKAN STEERING WHEEL DAN
OCULUS RIFT**

**DIMAS WIDDY PRATAMA
NRP 5112100144**

**Dosen Pembimbing
Darlis Herumurti, S.Kom., M.Kom.
Ridho Rahman H., S.Kom., M.Sc.**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016**

[Halaman ini sengaja dikosongkan]



FINAL PROJECT- KI141502

IMPLEMENTATION VIRTUAL REALITY OF CAR'S DRIVING LICENSE TEST USING STEERING WHEEL AND OCULUS RIFT

**DIMAS WIDDY PRATAMA
NRP 5112100144**

Advisor

Darlis Herumurti, S.Kom., M.Kom.

Ridho Rahman H., S.Kom., M.Sc.

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2016**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN SIMULASI UJIAN SURAT IZIN MENGEMUDI (SIM) MOBIL MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT

Tugas Akhir

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Rumpun Mata Kuliah Interaksi Grafika dan Seni
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

DIMAS WIDDY PRATAMA

NRP. 5112 100 144

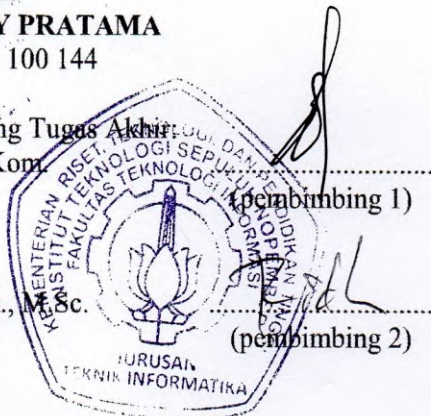
Disetujui oleh Dosen Pembimbing Tugas Akhir:

Darlis Herumurti, S.Kom., M.Kom.

NIP: 19761215 200312 1 001

Ridho Rahman Hariadi, S.Kom., M.Sc.

NIP: 19870213 201404 1 001



SURABAYA

JULI, 2016

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN SIMULASI UJIAN SURAT IZIN MENGEMUDI (SIM) MOBIL MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT

Nama Mahasiswa : Dimas Widdy Pratama
NRP : 51 12 100 144
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing I : Darlis Herumurti, S.Kom., M.Kom.
Dosen Pembimbing II : Ridho Rahman H., S.Kom., M.Sc.

ABSTRAK

Pada era global seperti ini, kendaraan bermotor sudah menjadi hal yang dipakai semua orang. Di Indonesia sendiri jalanan sering macet akibat banyaknya intensitas kendaraan bermotor yang ada di jalan. Namun apakah semua pengendara tersebut sudah terqualifikasi untuk mengendarainya? Di Indonesia sendiri dibutuhkan Surat Izin Mengemudi (SIM) agar dapat mengendarai kendaraan bermotor. Namun sering ditemui pengendara kendaraan tidak memiliki SIM. Terdapat indikasi bahwa kurangnya informasi mengenai ujian yang diujikan menjadi salah satu alasannya.

Diperlukan penghimpbauan kembali agar masyarakat Indonesia sadar akan pentingnya SIM itu sendiri. Apabila seseorang tidak pernah melakukan ujian SIM, maka tidak dapat dipastikan bahwa orang tersebut bisa mengendarai kendaraan bermotor dengan lancar. Salah satu solusinya adalah dibuatnya sebuah simulasi ujian praktek SIM yang dapat memberikan informasi kepada masyarakat tentang apa yang diujikan dalam ujian praktek SIM. Pengguna akan tahu bahwa dalam ujian praktek SIM A terdapat lima ujian, yaitu ujian maju mundur, ujian zig-zag, ujian parkir seri, ujian parkir parallel dan ujian menaiki tanjakan.

Dengan dikembangkannya aplikasi ini dapat membantu masyarakat yang belum meiliki surat izin mengemudi untuk mengetahui informasi ujian praktek sebelum mengambil ujian

sesungguhnya. . Dengan menggunakan Steering Wheel dan Oculus rift, maka akan memberikan pengalaman yang seperti sesungguhnya. Uji coba pada Tugas Akhir ini menggunakan skenario uji coba dengan metode kotak hitam dan kuesioner dari pengguna. Kuesioner ini menghasilkan nilai rata – rata 4 dari nilai maksimal yaitu 5.

Kata kunci: Simulasi, Surat Izin Mengemudi, Oculus Rift, Steering Wheel

IMPLEMENTATION VIRTUAL REALITY OF CAR'S DRIVING LICENSE TEST USING STEERING WHEEL AND OCULUS RIFT

Student Name : Dimas Widdy Pratama
NRP : 51 12 100 144
Major : Teknik Informatika FTIf-ITS
Advisor I : Darlis Herumurti, S.Kom., M.Kom.
Advisor II : Ridho Rahman H., S.Kom., M.Sc.

ABSTRACT

In this globalization era, motor vehicle is common for used by everyone. In Indonesia, it's normal to have traffic jam because the volume of the vehicle in the street. But, is everyone qualified for riding a motor vehicle? In Indonesia, it's required for rider to have a driving license called Surat Izin Mengemudi (SIM). But we can met rider who didn't have a driving license. For some reasone, they didn't have it because they didn't know what is being tested in driving license test.

There is a need for inform people so that they will realize the importance of driving license. If someone didn't take driving license test, they're did't qualified to riding a motor vehicle. One of the solution is to make a simulation about driving license test. People will know what is being tested in driving license test. The test is back and forth test, zig-zag test, usual parking test, parallel parking test and climb the hill test.

With the development of this application is expected to assist users in learning what is being tested in driving license test. Using Steering Wheel and Oculus Rift make the user feel the optimum experiecxce. This final project tested by using test scenarios with a black box methods and questionnaires from users. The final result of the quistionnaires is the application have 4 of 5 (maximum score) score.

Keywords: Simulation, Driving License, Oculus Rift, Steering Wheel

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji dan syukur, kehadiran Allah Subhanahu wa ta'ala yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Rancang Bangun Simulasi Ujian Surat Izin Mengemudi (SIM) Mobil menggunakan Steering Wheel dan Oculus Rift”.

Pengerjaan Tugas Akhir ini adalah momen bagi penulis untuk mengeluarkan seluruh kemampuan, hasrat, dan keinginan yang terpendam di dalam hati mulai dari masuk kuliah hingga lulus sekarang ini, lebih tepatnya di jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak. Melalui lembar ini, penulis ingin secara khusus menyampaikan ucapan terima kasih kepada:

1. Allah SWT yang telah melimpahkan rahmat, hidayah, dan inayah-Nya sehingga penulis mampu menyelesaikan Tugas Akhir dengan baik.
2. Junjungan kita Nabi Muhammad SAW yang telah menjadi inspirasi, contoh yang baik bagi penulis sehingga tetap termotivasi dalam mengerjakan Tugas Akhir.
3. Papa Mentari Hedy Swasono, Mama Wahyu Widati, Adik Ajeng Widdy Lestari, dan keluarga yang selalu memberikan dukungan dan semangat baik dalam bentuk motivasi ataupun materi. Dimana membuat penulis semangat dalam menyusun Tugas Akhir ini
4. Bapak Darlis Herumurti, S.Kom., M.Kom. dan Bapak Ridho Rahman H., S.Kom., M.Sc. yang telah bersedia untuk menjadi dosen pembimbing tugas akhir sehingga penulis dapat mengerjakan tugas akhir dengan arahan dan bimbingan yang baik dan jelas.

5. Teman-teman Mahasiswa Teknik Informatika 2012 yang telah berjuang bersama-sama selama menempuh pendidikan di Jurusan ini.
6. Pradipta Ghusti dan Sanindya Lesario selaku teman kontrakan yang selalu berada di sisi penulis dalam keadaan suka maupun duka.
7. Angga Saputra, Ikrom Aulia, Fahmy Thoriquil, Luthfi Soehadak, Fajar Setiawan dan Hafielludin yang telah membantu banyak dan menemani penulis dalam penyusunan Tugas Akhir ini.
8. Teman – teman himahot, Bimo Sigit, Yarjuna Rohmat, Regin Iqbal, Lukman Wahyu, Muhammad Baiquni, Aldrin Wiguna, Faris Makarim, Ashari Adhitama, Nicko Rahmadano, Ahmad Fathoriq, Soca Gumilar, Aditya Oktaviano yang selalu menghibur penulis jikalau terdapat masalah
9. Teman-teman seperjuangan di lab IGS (Interaksi, Grafika dan Seni), Wahyu Widyandana, Dimas Riskahadi, Tri Sutrisno, Aditya Ferza, Oshi Gusman, Rahma Fida, Pinasthika Cintya, Lubna Nur Aini, Ignatius Abraham, Devanda Tamba, Yohanes Aditya, Radhea Wicaksono, Christo Atan dan lainnya yang saling mendukung dalam memperjuangkan Tugas Akhir.

Penulis telah berusaha sebaik mungkin dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juli 2016

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	3
1.6 Metodologi	3
1.7 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA.....	7
2.1 Realitas Virtual.....	7
2.2 Unity	8
2.3 <i>Steering Wheel</i>	8
2.4 Ujian Tes Surat Izin Mengemudi.....	9
2.5 Oculus Rift.....	13
2.6 Oculus Utility	14
BAB III ANALISIS DAN PERANCANGAN.....	15
3.1 Analisis Perangkat Lunak.....	15
3.1.1 Deskripsi Umum Perangkat Lunak.....	15
3.1.2 Spesifikasi Kebutuhan Perangkat Lunak	16
3.1.3 Identifikasi Pengguna	17
3.2 Perancangan Perangkat Lunak.....	17
3.2.1 Perancangan Diagram Kasus Penggunaan.....	18
3.2.2 Definisi Kasus Penggunaan	19
3.2.3 Definisi Aktor	21

3.2.4	Perancangan Model.....	21
3.2.5	Arsitektur Umum Aplikasi.....	24
3.2.6	Perancangan Antarmuka Aplikasi.....	24
3.2.7	Perancangan Skenario Simulasi Ujian SIM.....	26
BAB IV IMPLEMENTASI.....		33
4.1	Lingkungan Implementasi	33
4.2	Implementasi Antarmuka.....	33
4.2.1	Implementasi Antarmuka Halaman Menu Utama....	33
4.2.2	Implementasi Antarmuka Halaman Simulasi Ujian.	37
4.3	Implementasi Integrasi Oculus Dengan Unity.....	51
4.4	Implementasi Sistem Kerja Mobil	52
4.5	Implementasi Skenario Simulasi Ujian SIM	57
4.5.1	Implementasi Skenario Ujian Maju Mundur	57
4.5.2	Implementasi Skenario Ujian <i>Zig – Zag</i>	60
4.5.3	Implementasi Skenario Ujian Parkir Seri.....	65
4.5.4	Implementasi Skenario Ujian Parkir Paralel.....	68
4.5.5	Implementasi Skenario Menaiki Tanjakan	71
4.6	Petunjuk Instalasi.....	74
BAB V PENGUJIAN DAN EVALUASI		77
5.1	Lingkungan Uji Coba	77
5.2	Skenario Pengujian	77
5.2.1	Pengujian Fungsionalitas	78
5.2.2	Pengujian Aplikasi Terhadap Pengguna	86
5.3	Evaluasi	87
5.3.1	Evaluasi Pengujian Fungsionalitas.....	88
5.3.2	Evaluasi Pengujian Aplikasi Terhadap Pengguna....	88
BAB VI KESIMPULAN DAN SARAN.....		91
6.1.	Kesimpulan.....	91
6.2.	Saran	92
DAFTAR PUSTAKA.....		93
LAMPIRAN		95
BIODATA PENULIS.....		101

DAFTAR GAMBAR

Gambar 2.1 Contoh Tampilan realitas Virtual	7
Gambar 2.2 Genius Speed Wheel 6MT	9
Gambar 2.3 Gambaran Ujian Maju Mundur	10
Gambar 2.4 Gambaran Ujian <i>Zig-Zag</i>	11
Gambar 2.5 Gambaran Ujian Parkir Seri	12
Gambar 2.6 Gambaran Ujian Paralel.....	12
Gambar 2.7 Gambaran Ujian Menaiki Tanjakan	13
Gambar 2.8 Oculus Rift.....	14
Gambar 3.1 Diagram Kasus Penggunaan	18
Gambar 3.2 Perancangan Model 3D Mobil Tampak Dalam	22
Gambar 3.3 Perancangan Arsitektur Aplikasi	24
Gambar 3.4 Rancangan Antarmuka Menu Utama	25
Gambar 3.5 Rancangan Antarmuka Ujian SIM	26
Gambar 3.6 Rancangan Skenario Tahap Pertama Ujian Maju Mundur	27
Gambar 3.7 Rancangan Skenario Tahap Kedua Ujian Maju Mundur	27
Gambar 3.8 Rancangan Skenario Tahap Pertama Ujian <i>Zig-Zag</i>	28
Gambar 3.9 Rancangan Skenario Tahap Kedua Ujian <i>Zig-Zag</i>	28
Gambar 3.10 Rancangan Skenario Ujian Parkir Seri	29
Gambar 3.11 Rancangan Skenario Ujian Parkir Paralel.....	30
Gambar 3.12 Rancangan Tahap Pertama Skenario Ujian Menaiki Tanjakan	30
Gambar 3.13 Rancangan Tahap Kedua Skenario Ujian Menaiki Tanjakan	31
Gambar 4.1 Antarmuka Halaman Menu Utama	34
Gambar 4.2 Pembuatan <i>Input Navigation</i>	36
Gambar 4.3 Perubahan Pada <i>EventSystem</i>	36
Gambar 4.4 Antarmuka Halaman Simulasi Ujian	37
Gambar 4.5 Tampilan <i>steer</i> pada mobil	38
Gambar 4.6 Tampilan Teks Navigasi	39
Gambar 4.7 Tampilan Kaca Spion Tengah	39

Gambar 4.8 Pembuatan <i>Texture Render</i> Kaca Spion Tengah	40
Gambar 4.9 Pembuatan Kamera Kaca Spion Tengah	41
Gambar 4.10 Pembuatan <i>Raw Image</i> Kaca Spion Tengah	41
Gambar 4.11 Tampilan Kaca Spion Kanan	42
Gambar 4.12 Pembuatan <i>Texture Render</i> Kaca Spion Kanan	43
Gambar 4.13 Pembuatan Kamera Kaca Spion Kanan	43
Gambar 4.14 Pembuatan <i>Raw Image</i> Kaca Spion Kanan.....	44
Gambar 4.15 Tampilan Kaca Spion Kiri	44
Gambar 4.16 Pembuatan <i>Texxure Render</i> Kaca Spion Kiri.....	45
Gambar 4.17 Pembuatan Kamera Kaca Spion Kiri	46
Gambar 4.18 Pembuatan <i>Raw Image</i> Kaca Spion Kiri.....	46
Gambar 4.19 Tampilan Layar Kamera Belakang	47
Gambar 4.20 Pembuatan <i>Texture Render</i> Layar Kamera Belakang	48
Gambar 4.21 Pembuatan Kamera Layar Kamera Belakang.....	48
Gambar 4.22 Pembuatan <i>Raw Image</i> Layar Kamera Belakang ..	49
Gambar 4.23 Penambah <i>Input</i> Untuk Perubahan <i>Gear</i>	50
Gambar 4.24 Tampilan Direktori yang Digunakan	51
Gambar 4.25 Tampilan <i>Player Setting</i>	52
Gambar 4.26 Tampilan <i>Box</i> dan <i>Wheel Collider</i>	53
Gambar 4.27 Implementasi Skenario Ujian Maju Mundur	58
Gambar 4.28 Implementasi Skenario Ujian <i>Zig-Zag</i>	61
Gambar 4.29 Implementasi Skenario Ujian Parkir Seri	66
Gambar 4.30 Implementasi Skenario Parkir Paralel.....	69
Gambar 4.31 Impelementasi Skenario Ujian Menaiki Tanjakan.	72
Gambar 4.32 Tampilan Pertama Instalasi Oculus Runtime.....	75
Gambar 4.33 Tampilan Oculus Runtime	75
Gambar 4.34 Tampilan Peringatan Kesehatan Pada Oculus Rift	76
Gambar 5.1 Kondisi Awal dan Akhir Pengujian Memilih Jenis Ujian SIM.....	78
Gambar 5.2 Kondisi Awal dan Akhir Pengujian Menjalankan Ujian SIM Secara Gagal.....	80
Gambar 5.3 Kondisi Awal dan Akhir Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Maju Mundur	81

Gambar 5.4 Kondisi Awal dan Akhir Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian <i>Zig-Zag</i>	82
Gambar 5.5 Kondisi Awal dan Akhir Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Parkir Seri	83
Gambar 5.6 Kondisi Awal dan Akhir Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Parkir Paralel.....	84
Gambar 5.7 Kondisi Awal dan Akhir Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Menaiki Tanjakan	85

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1 Karakteristik Pengguna.....	17
Tabel 3.2 Skenario Kasus Penggunaan	18
Tabel 3.3 Skenario Kasus Penggunaan Memilih jenis ujian SIM....	19
Tabel 3.4 Deskripsi pengguna.....	21
Tabel 3.5 Daftar <i>Asset</i> yang Digunakan	22
Tabel 5.1 Lingkungan Pengujian Perangkat Keras dan Lunak.....	77
Tabel 5.2 Pengujian Memilih Ujian SIM.....	79
Tabel 5.3 Tabel Pengujian Menjalankan Ujian SIM Secara Gagal .	80
Tabel 5.4 Tabel Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Maju Mundur	81
Tabel 5.5 Tabel Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian <i>Zig-Zag</i>	82
Tabel 5.6 Tabel Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Parkir Seri.....	83
Tabel 5.7 Tabel Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Parkir Paralel.....	84
Tabel 5.8 Tabel Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Menaiki Tanjakan.....	85
Tabel 5.9 Kuesioner Pengguna Simulasi Ujian SIM	86
Tabel 5.10 Tabel Hasil Evaluasi Pengujian Fungsionalitas.....	88
Tabel 5.11 Hasil Kuesioner Pengguna.....	89

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Kode Sumber Antarmuka Menu Utama	35
Kode Sumber 4.2 Kode Sumber <i>steer</i> pada mobil.....	38
Kode Sumber 4.3 Kode Sumber Perubahan Teks <i>Gear</i>	51
Kode Sumber 4.4 Kode Sumber Sistem Kerja Mobil.....	57
Kode Sumber 4.5 Kode Sumber Ujian Maju Mundur	60
Kode Sumber 4.6 Kode Sumber Skenario Ujian <i>Zig-Zag</i>	65
Kode Sumber 4.7 Kode Sumber Skenario Ujian Parkir Seri	68
Kode Sumber 4.8 Kode Sumber Skenario Ujian Parkir Paralel.....	71
Kode Sumber 4.9 Kode Sumber Skenario Ujian Menaiki Tanjakan	74

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kendaraan bermotor sudah menjadi kendaraan yang hampir semua orang memakainya. Pertumbuhan pengguna kendaraan bermotor sendiri sudah sangat meningkat di Indonesia. Hal ini memiliki dampak yang berpengaruh terhadap kehidupan sehari-hari. Adapun dampak yang baik maupun yang buruk. Dampak yang baik tentu dapat langsung dirasakan, salah satu contohnya adalah kita dapat mencapai tempat tujuan lebih cepat. Namun di antara dampak yang baik tersebut terdapat dampak yang buruk pula. Beberapa contoh dampak yang buruk adalah meningkatnya tingkat kecelakaan kendaraan bermotor dan macetnya jalanan dikarenakan intensitas kendaraan bermotor yang terus bertambah.

Pengguna kendaraan di Indonesia meningkat setiap harinya. Bahkan sekarang anak yang duduk di bangku SD pun sudah menggunakan kendaraan bermotor. Tentu sangat berbahaya apabila mereka belum cukup ahli dalam menggunakan kendaraan bermotor. Hal seperti kecelakaan pun bisa saja terjadi. Maka dari itu di Indonesia terdapat regulasi yaitu pengemudi bermotor harus memiliki SIM (Surat Izin Mengemudi) terlebih dahulu sebelum dapat menggunakan kendaraan bermotor. Namun dapat dilihat bahwa regulasi tersebut sering dilanggar oleh masyarakat. Seringkali mereka terlalu malas untuk mengambil ujian SIM, ataupun takut untuk mencoba ujian tersebut.

Tes ujian mengemudi bertujuan untuk mendapatkan *driving license* atau yang sering disebut Surat Izin Mengemudi di Indonesia. Namun banyak orang yang mengatakan bahwa SIM ini susah untuk didapatkan. Hal ini disebabkan oleh ujian praktek yang dapat dibilang tidak mudah. Pada ujian praktek, penguji dapat melihat kemahiran kita dalam mengemudikan kendaraan bermotor. Pengemudi harus menyelesaikan beberapa tes praktek menggunakan mobil seperti tes parkir parallel. Pengemudi sendiri apabila gagal

dalam ujian, mereka harus menunggu 1-2 minggu kedepan untuk bisa mengambil ujian ulang.

Simulasi yang dibuat ini bertujuan agar calon pengemudi dapat mencoba macam macam ujian praktek yang nantinya akan di ujikan. Dengan menggunakan *Steering Wheel* dan *Oculus Rift*, pengemudi diharapkan dapat merasakan suasana ujian yang nyata tanpa adanya rasa khawatir.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana bentuk ujian praktek SIM mobil sesuai dengan peraturan yang ada?
2. Bagaimana cara mengimplementasikan skenario ujian SIM ke dalam aplikasi simulasi?
3. Bagaimana cara mengintegrasikan *Steering Wheel* dan *Oculus Rift* pada *Unity3D*?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Menggunakan *Steering Wheel* Genius Speed Wheel 6 MT.
2. Perangkat VR yang di pakai adalah *Oculus Rift Development kit 2 (DK2)*.
3. Platform yang di gunakan adalah *Unity3D*.
4. Ujian pada simulasi sesuai dengan ujian praktek SIM A.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini antara lain:

1. Untuk mengetahui bentuk dari ujian praktik SIM mobil yang sesuai dengan peraturan yang ada.
2. Untuk dapat membuat *driving simulator* yang dapat di gunakan sebagai ujian tes SIM mobil.

3. Untuk dapat mengintegrasikan *Steering Wheel* dan Oculus Rift dengan Unity3D.

1.5 Manfaat

Manfaat dari hasil pembuatan Tugas Akhir ini adalah sebagai sarana berlatih untuk para calon pengemudi yang ingin mengambil tes SIM A.

1.6 Metodologi

Pembuatan Tugas Akhir dilakukan menggunakan metodologi sebagai berikut:

A. Studi literatur

Tahap studi literatur merupakan tahap pembelajaran dan pengumpulan informasi yang digunakan untuk mengimplementasikan Tugas Akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi dan pustaka, serta pemahaman dasar teori yang digunakan pada topik Tugas Akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

1. Realitas Virtual
2. Unity3D
3. Ujian SIM (Surat Izin Mengemudi) A
4. *Steering Wheel*
5. Oculus Rift

B. Perancangan perangkat lunak

Pada tahap ini diawali dengan melakukan analisis awal terhadap permasalahan utama yang muncul pada topik Tugas Akhir. Kemudian dilakukan perancangan perangkat lunak yang meliputi penentuan data yang digunakan dan proses-proses yang akan dilaksanakan. Langkah yang digunakan pada tahap ini adalah sebagai berikut:

1. Pencarian dan pendataan ujian surat izin mengemudi meliputi objek-objek apa saja yang dibuat dalam bentuk 3D.
2. Perancangan skenario ujian yang akan diujikan.
3. Perancangan visualisasi untuk membangun aplikasi realitas nyata.
4. Perancangan integrasi aplikasi dengan perangkat Oculus Rift.

C. Implementasi dan pembuatan sistem

Pada tahap ini dilakukan implementasi pemodelan objek-objek 3D untuk realitas virtual. Kemudian dilakukan integrasi aplikasi dengan perangkat Oculus Rift. Aplikasi ini dibangun dengan Unity.

D. Uji coba dan evaluasi

Pada tahap ini dilakukan uji coba dengan menggunakan beberapa macam kondisi untuk mencoba aplikasi bisa berjalan atau tidak. Uji fungsionalitas untuk mengetahui apakah aplikasi sudah memenuhi semua kebutuhan fungsional.

E. Penyusunan laporan Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil-hasil yang diperoleh selama pengerjaan Tugas Akhir.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini terdiri dari beberapa bab, yang dijelaskan sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan dan batasan permasalahan, tujuan dan manfaat pembuatan Tugas Akhir,

metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

BAB II TINJAUAN PUSTAKA

Bab ini membahas dasar pembuatan dan beberapa teori penunjang yang berhubungan dengan pokok pembahasan yang mendasari pembuatan Tugas Akhir ini.

BAB III ANALISIS DAN PERANCANGAN

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan, dan identifikasi pengguna. Kemudian membahas rancangan dari sistem yang dibuat meliputi rancangan skenario kasus penggunaan, arsitektur, data, dan antarmuka.

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembuatan realitas virtual ibadah haji dengan menggunakan Oculus Rift, dan antarmuka aplikasi.

BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi.

BAB VI PENUTUP

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran untuk pengembangan aplikasi selanjutnya.

[Halaman ini sengaja dikosongkan]

BAB II

TINJAUAN PUSTAKA

2.1 Realitas Virtual

Realitas virtual adalah sebuah teknologi dimana pengguna seolah – olah dapat merasakan berada di lingkungan yang telah disediakan. Dengan menggunakan realitas virtual, pengguna dapat berinteraksi secara tidak langsung ke dalam suasana tiga dimensi yang seolah nyata. Kebanyakan virtual realitas sekarang ini menggunakan alat penampil stereoskopik sebagai penampil visual. Karena realitas virtual ini adalah representatif dari lingkungan tiga dimensi yang tidak nyata, maka komputasi komputer dan visualisasi menjadi kunci kualitas realitas virtual itu sendiri.

Realitas virtual berbeda animasi ataupun video yang memiliki skenario tetap dimana apabila kita ulangi akan berasa tetap. Dengan realitas virtual, pengguna dapat melakukan berbagai interaksi dan dapat melihat dari berbagai perspektif. Realitas virtual ini memberikan fleksibilitas kepada penggunanya agar dapat menjelajah lingkungan virtual yang tidak nyata tersebut sesuai dengan keinginannya walaupun terbatas. Gambar 2.1 adalah contoh bentuk tampilan realitas virtual apabila ditampilkan di komputer [1].



Gambar 2.1 Contoh Tampilan realitas Virtual

2.2 Unity

Unity adalah ekosistem pengembangan game yang terintegrasi kuat dengan satu set lengkap alat intuitif dan alur kerja yang cepat untuk membuat 3D interaktif dan konten 2D. Unity merupakan *easy multiplatform publishing*. Unity memiliki tool *Asset* yang menyediakan *Asset* untuk di unduk secara gratis maupun berbayar. Terdapat pula *Unity Community* yang menyediakan tutorial secara gratis untuk semua pengguna unity. Fitur scripting yang disediakan, mendukung 3 bahasa pemrograman, yaitu JavaScript, C#, dan Boo. *Flexible and EasyMoving, rotating, dan scaling objects* hanya perlu sebaris kode. Begitu juga dengan *Duplicating, removing, dan changing properties*. Visual Properties Variables yang di definisikan dengan scripts ditampilkan pada Editor. Bisa digeser, di *drag and drop*, bisa memilih warna dengan *color picker*. Berbasis .NET. Artinya perjalanan program dilakukan dengan *Open Source .NET platform, Mono*. Serta mendukung pengembangan aplikasi Microsoft, SONY, Qualcomm, BlackBerry, Samsung, Nintendo, Oculus VR dan Intel [2].

2.3 Steering Wheel

Steering Wheel adalah suatu input untuk merealisasikan *game control*. Sesuai dengan namanya *game control* adalah perangkat *plug-in* yang di hubungkan sebuah perangkat sehingga memudahkan kita untuk berinteraksi ke dalam game tersebut. Contoh lain dari *game control* ini adalah *keyboard, mouse, gamepad, joystick, dan lain sebagainya*. Karena simulasi yang akan dibuat ini berbasis *driving simulator*, maka akan digunakan *steering wheel* Genius Speed Wheel 6 MT sebagai sebuah game controlnya. Agar calon pengemudi dapat merasakan seperti menggunakan mobil seperti aslinya. Gambar 2.2 merupakan contoh tampilan dari *steering wheel*.



Gambar 2.2 Genius Speed Wheel 6MT

Genius Speed Wheel 6MT merupakan jenis *steering wheel* yang memiliki bagian 2 bagian yaitu bagian atas dan bawah. Pada bagian atas terdapat setir yang digunakan untuk mengendalikan mobil ke kanan dan kiri. Terdapat juga tuas yang digunakan untuk mengganti gigi transmisi pada mobil. Pada bagian bawah terdapat dua buah pedal yang berfungsi untuk menaikkan *throttle* pada mobil dan mengerem mobil. Sangat disayangkan *steer* yang terdapat pada *steering wheel* ini hanya bisa melakukan putaran ke kanan dan ke kiri sebanyak 90°. Tidak seperti mobil sesungguhnya yang *steernya* dapat berputar sebanyak 360° [3].

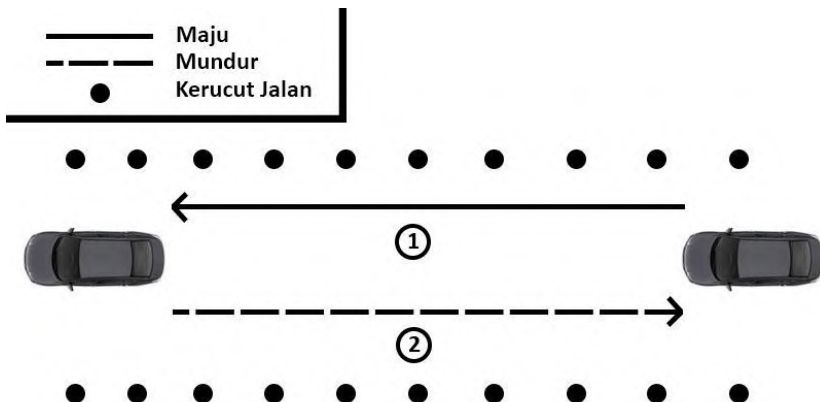
2.4 Ujian Tes Surat Izin Mengemudi

Di Indonesia, untuk mendapatkan Surat Izin Mengemudi (SIM) atau yang sering di sebut *driving license* di luar negeri, kita harus melalui beberapa tes yang telah di regulasi oleh Polisi Lalu Lintas (POLANTAS). Kita harus lulus dalam 2 tes yaitu tes tulis dan tes praktek. Dalam tes tulis kita akan di pertanyaan tertulis yang

menanyakan wawasan kita terhadap berkendara. Kita harus melampaui skor yang telah ditetapkan.

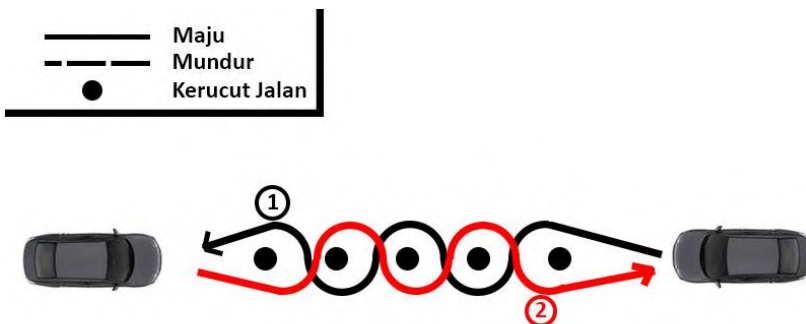
Tes kedua adalah ujian praktek. Dalam tes ini kita di uji kemampuan kita mengemudi secara langsung. Untuk tes ujian praktek SIM A, kita akan di uji menggunakan mobil yang telah di sediakan oleh POLANTAS. Kita akan disuruh meragakan 5 skenario yang telah di regulasi oleh mereka. Menurut Sumber yang didapat yang merupakan ujian SIM dari Gresik, kelima skenario tersebut adalah gerak maju lurus mundur lurus, *zig-zag* [4], parkir seri maju parkir seri mundur, parkir parallel dan melintasi tanjakan. Maka dari itu dalam pengerjaan Tugas Akhir ini penulis akan menirukan regulasi di atas untuk di terapkan dalam game yang akan di buat.

Ujian pertama adalah ujian maju mundur. Disini peserta ujian disuruh untuk mengemudikan mobil maju ke depan dan berhenti sebelum batas. Lalu pengemudi disuruh untuk mundur dan tidak melebihi batas yaitu posisi awal mobil tersebut. Gambar 2.3 akan menjelaskan ujian maju mundur.



Gambar 2.3 Gambaran Ujian Maju Mundur

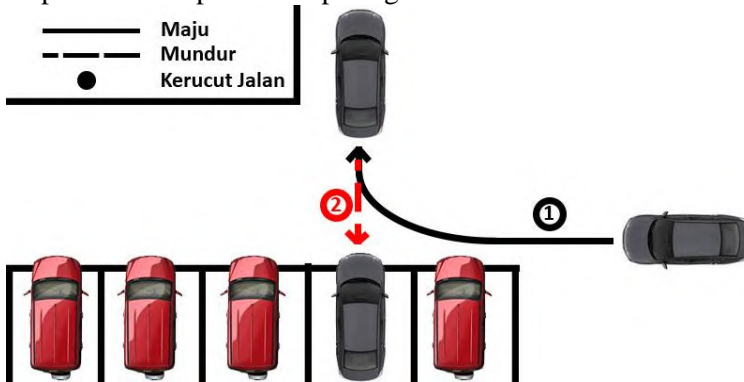
Ujian kedua adalah ujian *zig-zag* dimana peserta ujian harus mengendarai mobil melewati rintangan berupa kerucut jalan dengan cara *zig – zag*. Maksud dari *zig-zag* disini adalah melewatinya secara bergantian, awalnya melewati dari kanan, selanjutnya dari kiri. Begitu seterusnya hingga kerucut jalan yang ada habis. Kerucut jalan di letakkan secara lurus pada tengah jalan. Setelah itu, pengguna memutar balikkan mobil dan melewati kerucut secara bergantian lagi dan berhenti di tempat awal. Apabila mobil berhenti ke tempat awal maka ujian dianggap berhasil. Penjelasan ujian *zig-zag* dapat di lihat pada Gambar 2.4.



Gambar 2.4 Gambaran Ujian Zig-Zag

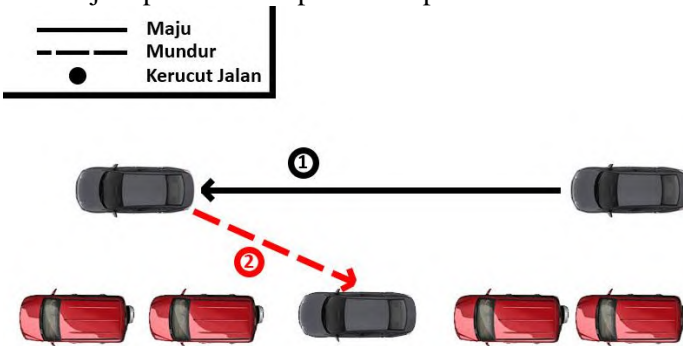
Ujian ketiga adalah ujian parkir seri dimana peserta disuruh untuk melakukan hal yang paling sering dilakukan pada saat mengendarai mobil yaitu melakukan parkir seri. Pada skenario ini terdapat sebuah tempat parkir seri. Pengguna disuruh untuk melakukan parkir secara seri dengan kepala mobil menghadap ke

depan ditempatkan yang telah disediakan diantara 2 mobil. Penjelasan ujian parkir seri dapat dilihat pada gambar 2.5.



Gambar 2.5 Gambaran Ujian Parkir Seri

Ujian keempat adalah ujian parkir paralel. kita disuruh untuk melakukan parkir paralel di antara mobil mobil yang sudah di atur secara vertikal. Skenario ini juga sering ditemukan pengendara mobil apabila kita ingin melakukan parkir mobil di jalan raya. Di jalan raya yang tidak memiliki lahan parkir kita dipaksa untuk harus memarkirkan mobil kita secara paralel. Pertama mobil mengambil posisi untuk melakukan parkir secara paralel. Lalu pengemudi harus mengemudikan mobil dan meletakkannya di antara mobil yang lain. Penjelasan ujian parkir seri dapat dilihat pada Gambar 2.6.



Gambar 2.6 Gambaran Ujian Paralel

Ujian terakhir adalah ujian menaiki tanjakan. Pada ujian ini peserta ujian disuruh untuk berhenti saat menaiki tanjakan. Skenario ini sering ditemukan apabila kita melakukan perjalanan ke gunung maupun jalan yang menanjak. Terkadang terjadi macet sehingga pengguna harus bisa mengerem agar tidak menabrak mobil yang berada di depannya. Pertama mobil akan maju menuju tanjakan dan menaikinya. Lalu saat mobil berada di tanjakan, mobil harus berhenti. Setelah berhenti beberapa saat, pengguna pun melanjutkan menaiki tanjakan dan berhenti di bawah tanjakan. Penjelasan ujian menaiki tanjakan dapat dilihat pada Gambar 2.7.



Gambar 2.7 Gambaran Ujian Menaiki Tanjakan

2.5 Oculus Rift

Oculus Rift adalah salah satu perangkat yang dikembangkan untuk memanipulasi pengalaman 3D. Sebuah headset ringan yang memungkinkan pengguna untuk masuk ke permainan dan melihat ke arah manapun. Oculus Rift sendiri adalah sebuah kacamata VR yang mempunyai dua layar untuk masing-masing mata. Dengan sudut pandang hingga 110 derajat, Oculus Rift akan menjadi kacamata yang bisa mewujudkan dunia maya seperti dunia nyata. Oculus Rift telah merilis versi komersialnya yang telah diperjual – belikan melalui website mereka. Versi ini lebih baru dan lebih canggih daripada versi sebelumnya yaitu Development Kit 1 (DK1)

dan Development Kit 2 (DK2) [5]. Gambar 2.8 merupakan contoh gambar dari oculus rift.



Gambar 2.8 Oculus Rift

2.6 Oculus Utility

Oculus utility adalah sebuah *package* yang apabila dibuka terdapat beberapa *asset* unity yang akan langsung masuk ke *project* yang sedang dibuka. *Asset* yang ada ada oculus utility ini berhubungan dengan pengembangan menggunakan oculus pada unity. Terdapat *asset* yang penting contohnya adalah OVRCameraRig. Ini adalah sebuah kamera pada unity yang memang dikhususkan untuk pengguna oculus. Jarak pandang yang sudah diatur dan beberapa fitur memang ditujukan lebih untuk aplikasi berbasis oculus [6]

BAB III

ANALISIS DAN PERANCANGAN

Bab ini menjelaskan tentang analisis dan perancangan simulasi ujian surat izin mengemudi mobil menggunakan Steering dan Oculus Rift . Pembahasan yang akan dilakukan meliputi analisis fitur yang dibutuhkan dan perancangan perangkat lunak.

3.1 Analisis Perangkat Lunak

Subbab ini menjelaskan tentang hasil analisis kebutuhan perangkat lunak serta arsitektur simulasi ujian surat izin mengemudi mobil menggunakan *Steering Wheel* dan Oculus Rift. Tiap-tiap subbab menjelaskan tentang deskripsi umum perangkat lunak, spesifikasi kebutuhan perangkat lunak, analisis aktor, arsitektur perangkat lunak, dan skenario kasus penggunaan.

3.1.1 Deskripsi Umum Perangkat Lunak

Pada Tugas Akhir ini dibuat sebuah simulasi ujian surat izin mengemudi menggunakan *Steering Wheel* dan Oculus Rift. Mengingat kendaraan di Indonesia sangatlah melonjak pesat tiap tahunnya, maka dirasa perlu untuk membuat simulasi ujian surat izin mengemudi ini. Sama dengan simulasi mengemudi yang lainnya, simulasi ini akan menggunakan *steering wheel* agar dapat memberikan suasana mengemudi yang nyata. Selain itu simulasi ini akan menggunakan Oculus Rift agar dapat memberikan visual yang nyata. Oculus Rift sendiri telah dibuktikan dapat memberikan pengalaman realitas virtual yang bagus.

Simulasi ini adalah simulasi ujian surat izin mengemudi mobil dimana akan membantu pengguna yang ingin mengambil ujian tersebut. Disini akan disimulasikan agar memiliki ujian yang sama dengan ujian yang diberikan oleh Polantas yaitu maju lurus mundur lurus, *zig – zag*, parkir seri, parkir paralel, dan melintasi tanjakan.

Pengguna akan diminta untuk memilih ujian mana yang ingin dia coba. Setelah memilih salah satu ujian, maka pengguna

akan berada di sebuah mobil, dimana mobil tersebut sudah berada di tengah – tengah lingkungan ujian. Mobil yang ada akan di lengkapi kaca spion kanan, kaca spion kiri dan kaca depan mobil. Lalu terdapat tulisan navigasi yang akan menuntun pengguna untuk menyelesaikan ujian tersebut. Apabila pengguna melakukan kesalahan seperti menabrak trotoar atau menabrak mobil lain, maka pengguna akan langsung gagal dan harus mencobanya dari awal. Apabila pengguna berhasil menyelesaikan ujiannya, maka pengguna dapat kembali ke menu utama dan dapat mencoba ujian yang lainnya.

3.1.2 Spesifikasi Kebutuhan Perangkat Lunak

Kebutuhan sistem yang dibuat ini melibatkan dua hal, yaitu kebutuhan fungsional dan non-fungsional. Dua kebutuhan tersebut akan dijelaskan lebih lanjut pada subbab berikut ini.

3.1.2.1 Kebutuhan Fungsional Perangkat Lunak

Berdasarkan deskripsi umum aplikasi, maka disimpulkan bahwa kebutuhan fungsional dari aplikasi ini adalah sebagai berikut.

1. Memilih jenis ujian SIM

Aplikasi dapat memiliki fungsionalitas memilih ujian yang ingin di coba saat itu. Sehingga pengguna tidak harus menyelesaikan ujian 1 untuk dapat mencoba ujian 2. Semua ujian dapat dijalankan kapanpun juga. Ujian pada aplikasi ini sendiri terbagi menjadi lima. Terdapat ujian maju mundur, *zig-zag*, parkir secara seri, parkir secara parallel dan menaiki dan menuruni tanjakan. Setelah ujian selesai (berhasil maupun gagal), pengguna dapat memilih untuk mencoba kembali ataupun ke menu untuk memilih ujian yang lainnya.

2. Menjalankan ujian SIM

Aplikasi dapat menampilkan *scene* dimana pengguna dapat menggunakan mobil yang seperti sesungguhnya. pengguna dapat mengendarainya dan melakukan serangkaian ujian yang di pilihnya. Serangkaian ujian tersebut adalah yaitu maju lurus

mundur lurus, *zig – zag*, parkir seri, parkir paralel, dan melintasi tanjakan.

3.1.2.2 Kebutuhan Non-Fungsional Perangkat Lunak

Terdapat beberapa kebutuhan non-fungsional yang apabila dipenuhi, dapat meningkatkan kualitas dari aplikasi ini. Berikut daftar kebutuhan non-fungsional:

1. Kebutuhan grafis

Grafis yang bagus dapat menambahkan nilai yang lebih terhadap suatu aplikasi. Kemiripan suatu objek pada aplikasi ini dapat membuat pengguna memiliki pengalaman yang maksimal.

2. Menciptakan suasana yang mirip dengan sesungguhnya

Pengguna akan mendapatkan pengalaman yang maksimal apabila dapat merasakan hal yang mirip dengan sesungguhnya. dalam konteks ini adalah suasana ujian SIM mobil.

3.1.3 Identifikasi Pengguna

Berdasarkan deskripsi umum diatas, maka dapat diketahui bahwa pengguna yang akan menggunakan aplikasi ini hanya satu orang, yaitu pengguna yang menjalankan aplikasi. Karakteristik pengguna tercantum dalam Tabel 3.1.

Tabel 3.1 Karakteristik Pengguna

Nama Aktor	Tugas	Hak Akses Aplikasi	Kemampuan yang harus dimiliki
Pengguna	Pihak luar yang mencoba aplikasi	Menjalankan aplikasi	Tidak ada

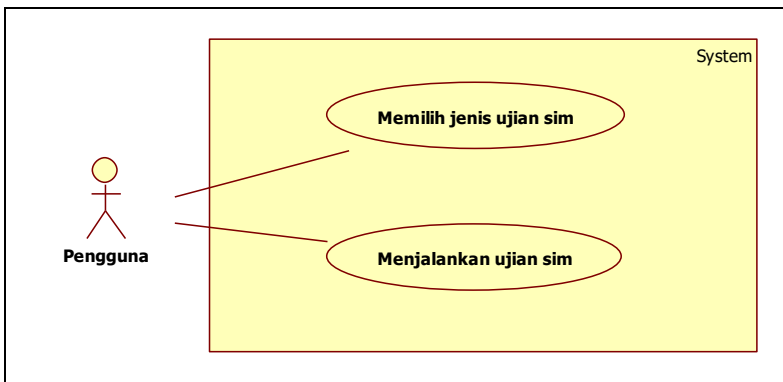
3.2 Perancangan Perangkat Lunak

Tahap perancangan dalam subbab ini dibagi menjadi beberapa bagian yaitu perancangan diagram kasus penggunaan, perancangan

skenario kasus penggunaan, perancangan data, perancangan model, perancangan informasi, perancangan antarmuka pengguna, dan perancangan kontrol aplikasi.

3.2.1 Perancangan Diagram Kasus Penggunaan

Dalam aplikasi Tugas Akhir ini, terdapat tiga kasus penggunaan yang ada yaitu memainkan piano virtual, memilih jenis musik, dan memilih tingkatan level permainan. Rancangan kasus penggunaan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Kasus Penggunaan

Penjelasan singkat dari masing-masing kasus penggunaan dapat dilihat pada Tabel 3.2.

Tabel 3.2 Skenario Kasus Penggunaan

No	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
1	UC-001	Memilih jenis ujian SIM	Pengguna dapat memilih serangkaian ujian SIM yang ada pada aplikasi.
2	UC-002	Menjalankan ujian SIM	Pengguna dapat menjalankan simulasi ujian SIM yang ada pada aplikasi.

3.2.2 Definisi Kasus Penggunaan

Subbab menjelaskan kasus penggunaan. Detail mengenai kasus penggunaan tersebut dapat dilihat pada subbab berikut ini.

3.2.2.1 Memilih jenis ujian SIM

Dalam kasus ini pengguna diharapkan dapat memilih satu dari lima skenario ujian yang telah disediakan. Sehingga pengguna dapat melatih kemampuan dalam setiap ujian. Spesifikasi kasus penggunaan memilih jenis ujian dapat dilihat pada Tabel 3.3.

Tabel 3.3 Skenario Kasus Penggunaan Memilih jenis ujian SIM

Nama Kasus Penggunaan	Memilih jenis ujian SIM
Kode	UC-001
Deskripsi	Kasus penggunaan agar actor dapat memilih jenis ujian SIM.
Aktor	Pengguna
Kondisi Awal	Pengguna sudah masuk ke aplikasi, dan sudah memakai oculus rift dan <i>steering wheel</i> .
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aplikasi akan menampilkan button pilihan ujian SIM 2. Pengguna menggerakkan steer pada <i>steering wheel</i> ke kanan untuk menggerakkan hover tombol ke ujian berikutnya <ol style="list-style-type: none"> A1. Pengguna menggerakkan steer pada <i>steering wheel</i> ke kanan untuk menggerakkan hover tombol ke ujian sebelumnya 3. Pengguna mamilih ujian mana yang ingin dijalankan 4. Pengguna menekan tombol x pada <i>steering wheel</i> sebagai tombol pemilih 5. Aplikasi menampilkan <i>scene</i> ujian sim yang di pilih
Alur Kejadian	A1. Pengguna menggerakkan steer pada <i>steering wheel</i> ke kiri untuk menggerakkan hover ke

Nama Kasus Penggunaan	Memilih jenis ujian SIM
Alternatif	ujian sebelumnya 1. Steer pada <i>steering wheel</i> digerakkan ke kiri untuk menggerakkan hover tombol ke ujian sebelumnya 2. Kembali ke alur kejadian normal nomor 3

3.2.2.2 Menjalankan ujian SIM

Dalam kasus ini pengguna diharapkan dapat melakukan simulasi skenario ujian yang telah dipilih. Terdapat lima skenario ujian yaitu maju lurus mundur lurus, *zig – zag*, parkir seri, parkir paralel, dan melintasi tanjakan. Spesifikasi kasus penggunaan menjalankan ujian SIM dapat dilihat pada Tabel 3.4.

Nama Kasus Penggunaan	Menjalankan ujian SIM
Kode	UC-002
Deskripsi	Kasus penggunaan agar aktor dapat menjalankan ujian SIM
Aktor	Pengguna
Kondisi Awal	Pengguna sudah masuk ke salah satu skenario ujian SIM, dan sudah memakai oculus rift dan <i>steering wheel</i> .
Alur Normal	1. Aplikasi <i>scene</i> dengan suasana ujian SIM yang dipilih. 2. Aplikasi menampilkan teks navigasi. 3. Pengguna melihat teks navigasi agar mengerti perintah skenario. 4. Pengguna menginjak pedal kanan pada <i>steering wheel</i> . A1. Pengguna menginjak pedal kiri pada <i>steering wheel</i> . A2. Pengguna menggerakkan <i>steer</i> pada <i>steering wheel</i> ke kanan. A3. Pengguna menggerakkan <i>steer</i> pada <i>steering wheel</i> ke kiri.

	5. Mobil yang dikendarai pengguna pada aplikasi menambah kecepatan. 6. Pengguna berhasil menjalankan skenario pada simulasi. A4. Pengguna gagal menjalankan skenario pada simulasi.
Alur Alternatif	A1. Pengguna menginjak pedal kiri pada <i>steering wheel</i> .. 1. Mobil yang dikendarai pengguna pada aplikasi mengurangi kecepatan . 2. Kembali ke alur kejadian normal nomor 6. A2. Pengguna menggerakkan <i>steer</i> pada <i>steering wheel</i> ke kanan. 1. Mobil yang dikendarai pengguna pada aplikasi belok ke kanan. 2. Kembali ke alur kejadian normal nomor 6. A3. Pengguna menggerakkan <i>steer</i> pada <i>steering wheel</i> ke kiri. 1. Mobil yang dikendarai pengguna pada aplikasi 2. Kembali ke alur kejadian normal nomor 6.

3.2.3 Definisi Aktor

Aktor yang terdapat dalam aplikasi simulasi ujian SIM terlihat pada Tabel 3.5

Tabel 3.4 Deskripsi pengguna

No	Nama	Deskripsi
1	Pengguna	Aktor yang bertugas untuk menggunakan dan menjalankan semua fungsionalitas pada aplikasi ini

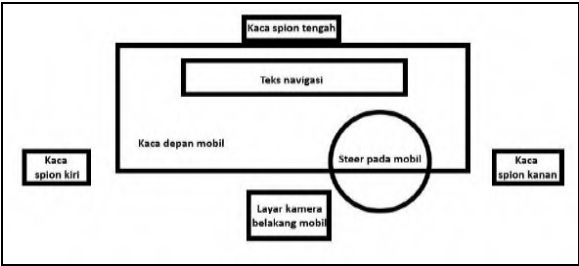
3.2.4 Perancangan Model

Perancangan model pada aplikasi simulasi ujian SIM ini diambil dari berbagai sumber seperti Asset Store pada Unity ataupun

berbagai website penyedia model 3D. Detik rancangan model ditampilkan pada subbab berikutnya.

3.2.4.1 Perancangan Model Interior Mobil

Terdapat beberapa model yang dibutuhkan. Yaitu kaca spion kanan kiri tengah, dan juga *steer*. Rancangan model interior pada aplikasi simulasi ujian SIM dapat dilihat pada Gambar 3.2.





Gambar 3.2 Perancangan Model 3D Mobil Tampak Dalam

3.2.4.2 Daftar Asset

Adapun beberapa *asset* yang digunakan dalam pembuatan aplikasi simulasi ini. Namun Asset yang digunakan dalam pembuatan tugas akhir ini bukanlah buatan sendiri namun merupakan *asset* gratis yang legal dan boleh digunakan. Beberapa *asset* yang digunakan disebutkan dalam Tabel 3.6 berikut sumbernya

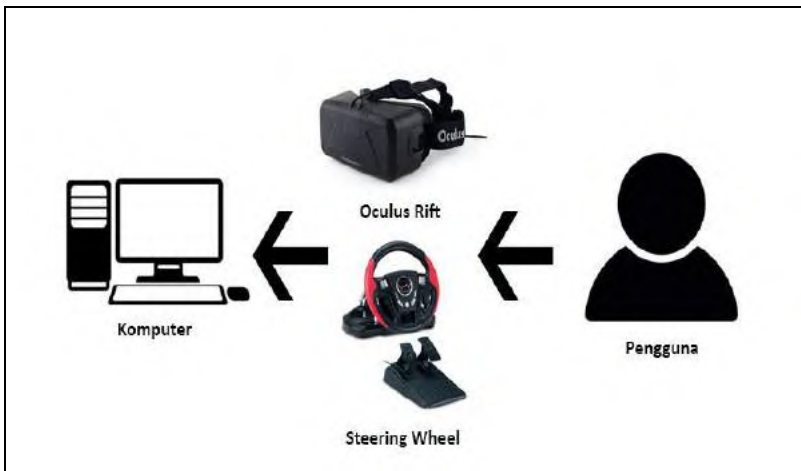
Tabel 3.5 Daftar *Asset* yang Digunakan

No	Gambar	Sumber
1		http://tf3dm.com/3d-model/audi-s3-2011-29265.html

2		http://tf3dm.com/3d-model/street-system-v10-48448.html
3		http://www.turbosquid.com/FullPreview/Index.cfm/ID/591674
4		http://www.turbosquid.com/FullPreview/Index.cfm/ID/342348
5		https://www.assetstore.unity3d.com/en/#!/content/47386

3.2.5 Arsitektur Umum Aplikasi

Arsitektur umum pada aplikasi simulasi ujian SIM ini menggunakan beberapa perangkat tambahan. Antara lain adalah *Steering Wheel* dan Oculus Rift. Implementasi aplikasi ini dibuat menggunakan Unity. Digunakan pula Blender untuk membuat dan mengedit objek 3D model yang dipakai di aplikasi ini. Arsitektur secara umum aplikasi dapat dilihat pada Gambar 3.3.



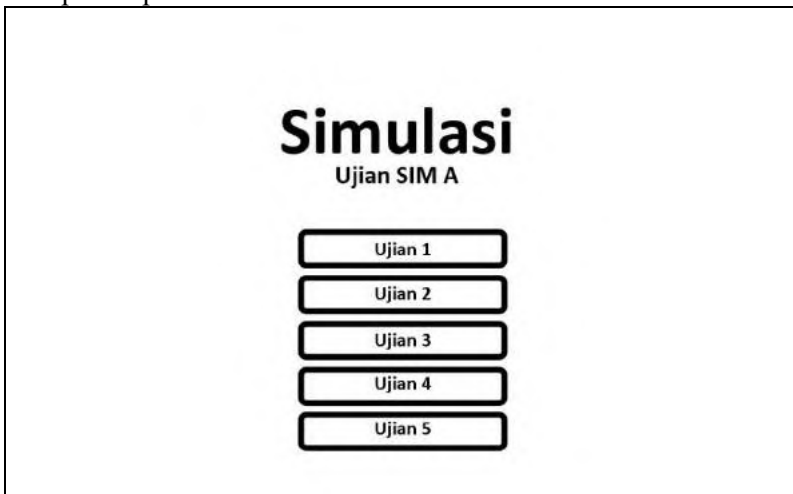
Gambar 3.3 Perancangan Arsitektur Aplikasi

3.2.6 Perancangan Antarmuka Aplikasi

Rancangan antarmuka aplikasi diperlukan untuk memberikan gambaran umum kepada pengguna bagaimana sistem yang ada dalam aplikasi ini berinteraksi dengan pengguna. Selain itu, rancangan ini juga memberikan gambaran bagi pengguna apakah tampilan yang sudah disediakan oleh aplikasi mudah untuk dipahami dan digunakan, sehingga akan muncul kesan *user experience* yang baik dan mudah. Antarmuka yang akan dibuat diharapkan bisa memberikan pengalaman seperti di atas. Pada bagian ini dijelaskan mengenai perancangan antarmuka pada aplikasi simulasi ujian SIM.

3.2.6.1 Antarmuka Menu Utama

Halaman antarmuka menu utama pada aplikasi ini muncul pada saat pertama kali pengguna membuka aplikasi ini. Pada antarmuka menu utama terdapat pilihan ujian yang direpresentasikan oleh beberapa tombol. Terdapat 5 tombol yang tiap tombolnya merepresentasikan satu ujian. Rancangan antarmuka menu utama ditampilkan pada Gambar 3.4.



Gambar 3.4 Rancangan Antarmuka Menu Utama

3.2.6.2 Antarmuka ujian SIM

Halaman antarmuka ujian SIM ini muncul ketika pengguna telah memilih salah satu dari berbagai rangkaian ujian yang ada pada antarmuka menu utama. Ketika berada di antarmuka ini pengguna akan berada di dalam mobil pada tempat duduk pengemudi. Terdapat beberapa tampilan pembantu yang ada pada antarmuka ini. Terdapat kaca spion tengah dan teks navigasi pada bagian atas. Kaca spion tengah berfungsi untuk menampilkan bagian belakang mobil, sementara teks navigasi berfungsi untuk memberi petunjuk pengguna agar dapat menyelesaikan skenario ujian SIM. Terdapat

kaca spion kanan pada bagian kanan yang berfungsi untuk memberi tampilan bagian kanan belakang pada mobil. Terdapat kaca spion kiri pada bagian kiri yang berfungsi untuk memberi tampilan bagian kiri belakang pada mobil. Lalu terdapat layar kamera belakang mobil pada bagian tengah yang berfungsi untuk memberikan tampilan belakang bawah mobil. Yang terakhir terdapat *steering wheel* yang berada di depan pengguna yang berguna untuk menunjukkan state *steering wheel* sekarang. Tampilan rancangan antarmuka ujian SIM ditampilkan pada Gambar 3.5.



Gambar 3.5 Rancangan Antarmuka Ujian SIM

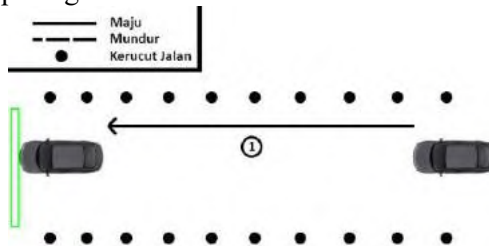
3.2.7 Perancangan Skenario Simulasi Ujian SIM

Pada subbab ini akan memberi gambaran mengenai skenario pada setiap ujian. Terdapat 5 skenario ujian pada aplikasi ini. Ujian maju – mundur, zig – zag, parkir secara ‘seri’, parkir secara ‘parallel’, dan menaiki tanjakan. Untuk merancang skenario pada aplikasi simulasi ini, digunakan *Collider* yang berfungsi sebagai

pemacu agar even lainnya terjadi Untuk penjelasan lebih detailnya dapat dilihat pada subbab berikut ini.

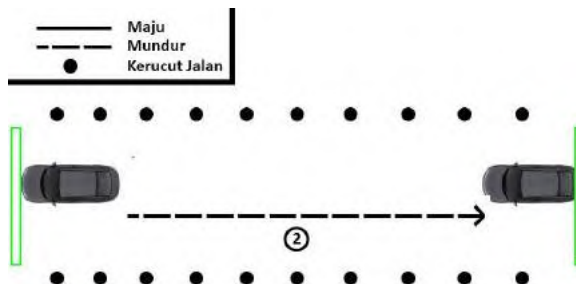
3.2.7.1 Rancangan Skenario Ujian Maju Mundur

Pada awal skenario, terdapat *collider* yang di letakkan di ujung jalan. Aplikasi akan menyuruh pengguna untuk mengendarai mobil menuju ujung jalan, namun tidak melewatinya. Penjelasan tahap ini dapat dilihat pada gambar 3.6



Gambar 3.6 Rancangan Skenario Tahap Pertama Ujian Maju Mundur

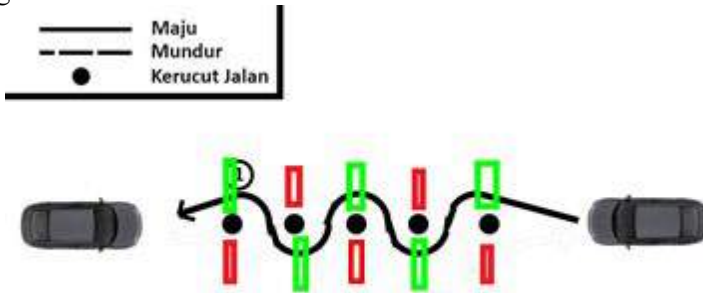
Setelah menabrak *collider* dan mobil dalam keadaan berhenti, maka akan muncul *collider* di tempat awal mobil sebelum di kemudikan. Lalu aplikasi akan menyuruh pengemudi untuk mengendarai mobil secara mundur kembali ke tempat awal. Apabila mobil menyentuh *collider* dan mobil dalam keadaan berhenti. Maka skenario dapat dikatakan berhasil. Penjelasan tahap kedua dapat dilihat pada gambar 3.7.



Gambar 3.7 Rancangan Skenario Tahap Kedua Ujian Maju Mundur

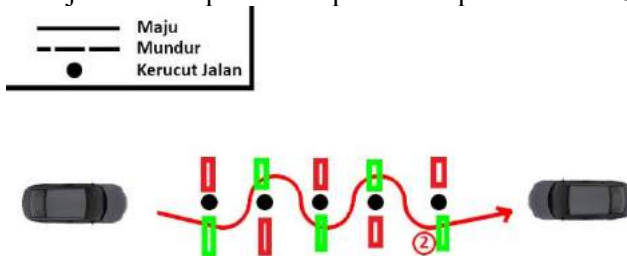
3.2.7.2 Rancangan Skenario Ujian Zig-Zag

Pada awal sekanrio ujian ini, terdapat beberapa kerucut jalan yang berperan sebagai rintangan. Lalu diletakkan *collider* sesuai dengan Gambar 3.8



Gambar 3.8 Rancangan Skenario Tahap Pertama Ujian Zig-Zag

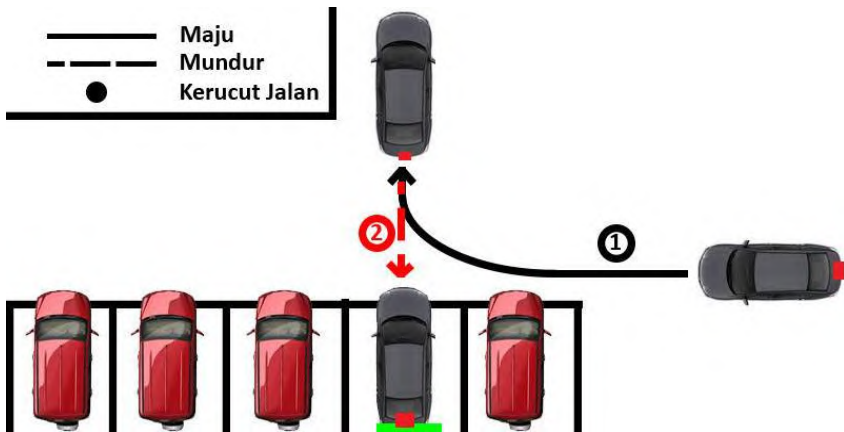
Pada gambar di atas, terdapat *collider* berwarna merah dan berwarna hijau. *Collider* berwarna hijau berarti boleh dilewati, sedangkan yang berwarna merah tidak dapat di lewati. Lalu setelah mobil berada di ujung jalan, maka *collider* tadi hancur dan berganti *collider* lain yang sifatnya berbeda. Apabila tadinya berwarna merah, maka akan di ganti berwarna hijau. Begitu juga sebaliknya. Setelah mobil berada di tempat awal, maka skenario dapat dikatakan berhasil. Penjelasan tahap kedua dapat dilihat pada Gambar 3.9.



Gambar 3.9 Rancangan Skenario Tahap Kedua Ujian Zig-Zag

3.2.7.3 Rancangan Skenario Ujian Parkir Seri

Pada ujian ini, pengemudi disuruh untuk melakukan parkir secara seri di antara dua mobil. Terdapat *collider* pada belakang tempat parkir dan belakang mobil. Penjelasan rancangan ujian parkir seri dapat dilihat pada gambar 3.10.

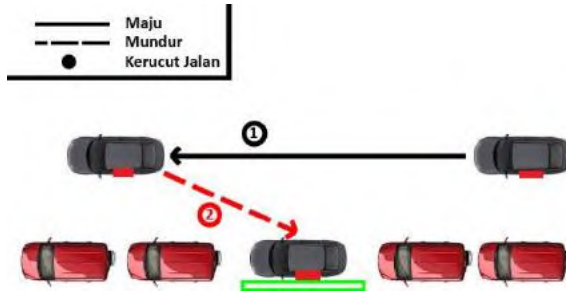


Gambar 3.10 Rancangan Skenario Ujian Parkir Seri

Pada gambar diatas tedapat *collider* bwwarna hijau dan berwarna biru. *Collider* berwarna hijau adalah collider yang berada pada ujung tempat parkir. Lalu *collider* berwarna merah adalah *collider* yang berada pada belakang mobil. Apabila mobil melakukan parkir seri dengan benar, maka seharusnya *collider* merah dan hijau bertabrakan dan skenario dianggap berhasil.

3.2.7.4 Rancangan Skenario Ujian Parkir Paralel

Pada ujian ini, sangatlah mirip dengan ujian sebelumnya. Hanya saja letak *collidernya* yang berbeda. Pada ujian ini *collider* di letakkan pada ujung tempat parkir dan bagian samping mobil. Penjelasan rancangan ujian parkir parallel dapat di lihat pada Gambar 3.11.

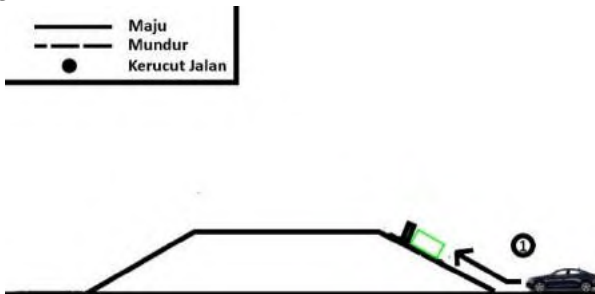


Gambar 3.11 Rancangan Skenario Ujian Parkir Paralel

Pada Gambar di atas terdapat *collider* berwarna merah dan hijau. *Collider* hijau di letakkan di ujung tempat parkir. Lalu terdapat *collider* berwarna merah yang di letakkan di samping mobil. Sehingga apabila kedua *collider* tersebut bersentuhan, maka mobil akan parkir secara parallel dan skenario dapat dikatakan berhasil.

3.2.7.5 Rancangan Skenario Ujian Menaiki Tanjakan

Pada ujian ini pengguna disuruh untuk berhenti saat menaiki tanjakan. Pada awalnya akan ada penghalang pada tanjakan. Sebelum penghalang tersebut terdapat *collider*. Penjelasan tahap pertama perancangan ujian menaiki tanjakan dapat dijelaskan pada Gambar 3.12.



Gambar 3.12 Rancangan Tahap Pertama Skenario Ujian Menaiki Tanjakan

Setelah mobil yang dikendarai oleh pengguna berada dalam *collider* dan berhenti dan tidak menabrak penghalang, maka *countdown* waktu selama lima detik. Setelah itu muncullah *collider* baru yang berada pada tempat awal mobil berada. *Collider* ini di bikin apabila mobil tidak berhasil menaiki tanjakan maka mobil akan dianggap gagal. Setelah itu pengemudi dapat menaiki tanjakan dan menuruni tanjakan. Penjelasan tahap kedua dapat dilihat pada Gambar 3.13.



Gambar 3.13 Rancangan Tahap Kedua Skenario Ujian Menaiki Tanjakan

Collider hijau akan muncul setelah *countdown* lima detik berakhir. Sehingga apabila mobil gagal menanjak maka skenario dianggap gagal. Skenario akan dianggap berhasil apabila mobil berhasil menaiki dan menuruni tanjakan yang tersedia.

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan perangkat lunak. Cakupan implementasi dari perancangan perangkat lunak tersebut meliputi proses penerapan dan pengimplementasian algoritma dan antarmuka.

4.1 Lingkungan Implementasi

Lingkungan implementasi Tugas Akhir dijelaskan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Keras dan Lunak

Perangkat Keras	Prosesor : - Intel® Core™ i5-4460 Processor CPU@3.20GHz Memori : - 8 GB Kartu Grafis : - Nvidia GeForce GTX 960
Perangkat Lunak	Sistem Operasi : - Microsoft Windows 10 Pro 64-bit Perangkat Pengembang : - Unity 5.3.4f1 - Visual Studio Ultimate 2012

4.2 Implementasi Antarmuka

Subbab ini membahas mengenai implementasi antarmuka pengguna yang meliputi implementasi antarmuka halaman menu utama, dan antarmuka halaman simulasi ujian sim.

4.2.1 Implementasi Antarmuka Halaman Menu Utama

Tampilan pertama yang muncul pada aplikasi ini adalah antarmuka halaman menu utama dan di implementasikan di MainMenu.unity. Pada antarmuka halaman menu utama terdapat logo yang merupakan tulisan ‘SIMULASI UJIAN SIM A’. Lalu terdapat 6 tombol tepat berada di bawah logo. Tombol satu sampai

ke lima adalah tombol yang merepresentasikan setiap ujian yang ada pada aplikasi ini. Sedangkan tombol ke 6 adalah tombol exit yang berfungsi untuk keluar dalam aplikasi ini. Tampilan antarmuka halaman menu utama dapat dilihat pada Gambar 4.1.



Gambar 4.1 Antarmuka Halaman Menu Utama

Pada setiap tombol yang ada pada antarmuka halaman menu utama terdapat kode yang membuat aplikasi berpindah *scene* sesuai dengan tombol yang dipilih. Tombol ‘Ujian 1’ sampai ‘Ujian 5’ akan membuat *scene* berpindah ke simulasi ujian masing – masing. Sedangkan tombol ‘Exit’ bertugas untuk keluar aplikasi. Implementasi halaman menu utama dapat dijelaskan pada Kode Sumber 4.1 yang berada pada kelas MainMenuButton.

```
using UnityEngine;
using System.Collections;

public class MainMenuButton : MonoBehaviour {
    public void Button1()
    {
        Application.LoadLevel("Ujian 1");
    }
    public void Button2()
```



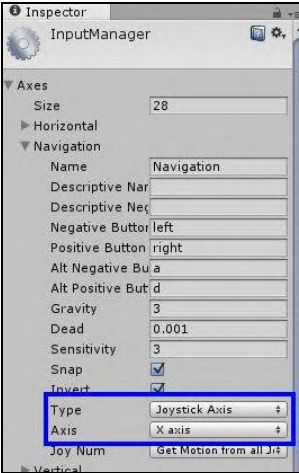
```

{
    Application.LoadLevel("Ujian 2");
}
public void Button3()
{
    Application.LoadLevel("Ujian 3");
}
public void Button4()
{
    Application.LoadLevel("Ujian 4");
}
public void Button5()
{
    Application.LoadLevel("Ujian 5");
}
public void ExitButton()
{
    Application.Quit();
}
}

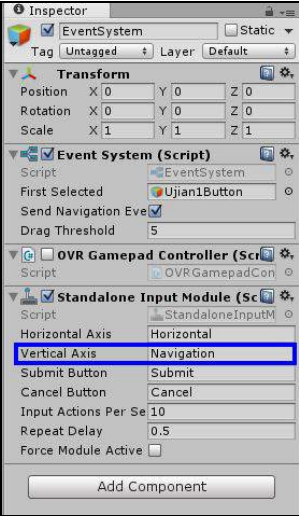
```

Kode Sumber 4.1 Kode Sumber Antarmuka Menu Utama

Pada antarmuka halaman menu utama, pemindahan antar tombol akan terjadi apabila kita menggerakkan *steering wheel* ke kanan atau ke kiri. Prosesnya sama dengan menggerakkan mobil ke kanan dan ke kiri. Maka dari itu kita perlu membuat input baru pada input manager [7] yang memiliki *type* berupa *Joystick Axis* karena inputan kita berupa *joystick* yaitu *steering wheel* dan *Axis* berupa *X Axis* karena belok ke kanan dan ke kiri merupakan *X axis*. Pada implementasi ini input baru ini di beri nama *Navigation*. Lalu *vertical axis* pada *standalone input module (Script)* yang berada pada *EventSystem* dirubah ke inputan yang baru dibuat yaitu *Navigation*. Sehingga pemindahan antar button bisa menggunakan *steering wheel*. Pembuatan input baru pada *input manager* akan dijelaskan pada Gambar 4.2. Lalu perubahan yang ada pada *EventSystem* akan dijelaskan di Gambar 4.3.



Gambar 4.2 Pembuatan *Input Navigation*



Gambar 4.3 Perubahan Pada *EventSystem*

4.2.2 Implementasi Antarmuka Halaman Simulasi Ujian

Setelah antarmuka halaman menu utama, maka antarmuka selanjutnya adalah antarmuka halaman simulasi ujian. Antarmuka ini akan muncul pada setiap skenario simulasi ujian. Karena pada setiap antarmuka ujian memiliki tampilan yang sama, maka semuanya akan dijelaskan apada subbab ini. Tampilan antarmuka halaman simulasi ujian akan diperlihatkan pada Gambar 4.4.



Gambar 4.4 Antarmuka Halaman Simulasi Ujian

Pada antarmuka ini terdapat enam elemen yang penting pada antarmuka ini. *Steer* pada mobil, teks navigasi, kaca spion tengah, kaca spion kiri, kaca spion kanan, layar kamera belakang mobil. *Steer* pada mobil merepresentasikan gerak alat *steering wheel*. Teks navigasi berisi teks yang bertujuan untuk memberitahu pengguna apa yang harus di lakukan. Kaca spion kiri berfungsi sebagai kaca spion di sebelah kiri pada mobil. Kaca spion kanan berfungsi sebagai kaca spion di sebelah kanan pada mobil. Kaca spion tengah berfungsi sebagai kaca spion di sebelah tengah atas pada mobil. Layar kamera belakang mobil berfungsi sebagai layar yang menunjukkan kamera belakang dan penunjuk *gear* pada mobil. Semua elemen ini ada sebagai pembantu saat melakukan simulasi

ujian. Dan memang enam elemen ini terdapat pada mobil pada umumnya.

4.2.2.1 Implementasi *steer* pada mobil

Pada implementasi aplikasi ini menggunakan alat *steering wheel* Genius Speed Wheel 6 MT. Maka dari itu, seharusnya *steer* pada mobil dalam aplikasi harus mengikuti pergerakan dari alat *steering wheel*. Tampilan *steer* pada mobil dapat dilihat pada Gambar 4.5.



Gambar 4.5 Tampilan *steer* pada mobil

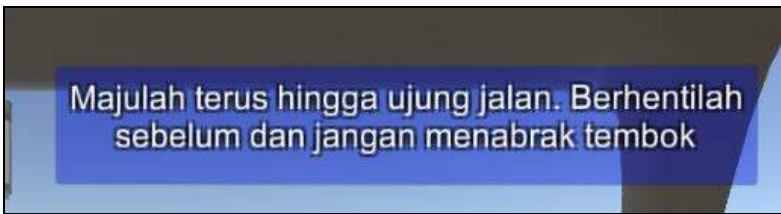
Maka dari itu untuk mengikuti pergerakan dari alat *steering wheel*, diperlukan kode sumber yang di taruh di *steer* pada mobil. Implementasi *steer* pada mobil dapat dijelaskan pada Kode Sumber 4.2 yang berada pada kelas CarControl.

```
void UpdateSteerAngle()
{
    float steer = Input.GetAxis("Horizontal") * -90;
    float posy = car.transform.eulerAngles.y;
    float posx = car.transform.eulerAngles.x;
    SteerWheel.transform.eulerAngles = new
    Vector3(posx, posy, steer);
}
```

Kode Sumber 4.2 Kode Sumber *steer* pada mobil

4.2.2.2 Implementasi Teks Navigasi

Teks navigasi ini bertujuan untuk memberi perintah kepada pengguna. Teks ini berisi apa yang harus dilakukan oleh pengguna agar dia dapat menyelesaikan simulasi ujian tersebut. Teks ini akan berubah apabila pengguna telah berhasil melakukan perintah sebelumnya. Apabila pengguna berhasil melakukan apa yang ditulis pada teks navigasi terakhir pada simulasi ujian tersebut, maka pengguna dikatakan berhasil menjalankan simulasi ujian tersebut. Tampilan teks navigasi dapat dilihat pada Gambar 4.6.



Gambar 4.6 Tampilan Teks Navigasi

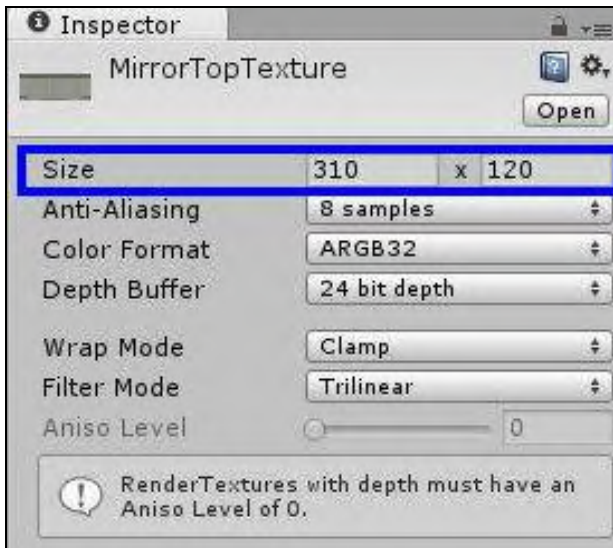
4.2.2.3 Implementasi Kaca Spion Tengah

Kaca spion tengah bertujuan untuk merepresentasikan kaca spion tengah atas yang berada pada mobil. Dengan adanya kaca spion tengah ini pengguna akan lebih mudah melihat bagian belakang mobil. Tampilan kaca spion tengah dapat dilihat pada Gambar 4.7



Gambar 4.7 Tampilan Kaca Spion Tengah

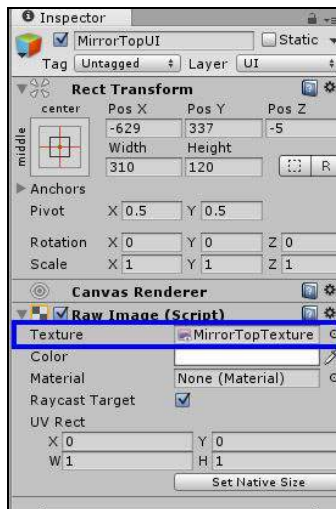
Kaca spion tengah ini di implementasikan dengan meletakkan kamera yang menghadap ke belakang sehingga kamera itu memperlihatkan seolah olah kaca yang menghadap kebelakang. lalu dengan *raw image* pada *canvas UI* kita dapat menampilkan gambar yang kamera tersebut perlihatkan. Pada awalnya kita membuat *render texture* baru yang memiliki ukuran sama dengan *raw image*-nya. Pembuatan *texture render* baru dapat dilihat pada Gambar 4.8. Lalu dibuatlah kamera baru yang diletakan menghadap belakang dan diletakkan di tengah atas pada dalam mobil. Lalu kita mengganti value *target texture* menjadi *texture render* yang tadi kita buat. Pembuatan kamera dapat dilihat pada Gambar 4.9. Lalu dibuatlah *raw image* baru yang diletakkan sama dengan kamera yang tadi dibuat. Value *texture* pada *raw image* akan diganti dengan *render texture* yang tadi telah dibuat. Pembuatan *raw image* dapat dilihat pada Gambar 4.10.



Gambar 4.8 Pembuatan *Texture Render* Kaca Spion Tengah



Gambar 4.9 Pembuatan Kamera Kaca Spion Tengah



Gambar 4.10 Pembuatan *Raw Image* Kaca Spion Tengah

4.2.2.4 Implementasi Kaca Spion Kanan

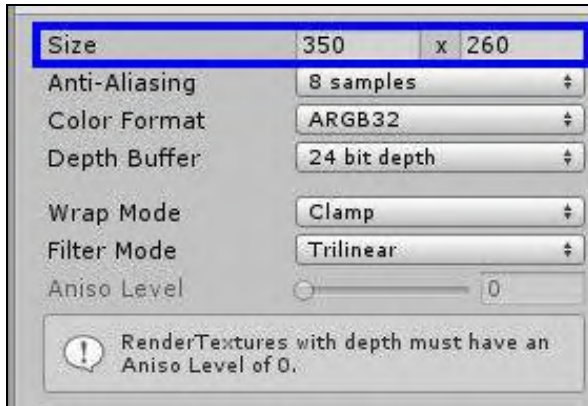
Kaca spion kanan bertujuan untuk merepresentasikan kaca spion kanan yang berada pada mobil. Dengan adanya kaca spion kanan ini pengguna akan lebih mudah melihat bagian belakang kanan mobil. Tampilan kaca spion kiri dapat dilihat pada Gambar 4.11.



Gambar 4.11 Tampilan Kaca Spion Kanan

Kaca spion kanan ini di implementasikan dengan meletakkan kamera yang menghadap ke belakang sehingga kamera itu memperlihatkan seolah olah kaca yang menghadap kebelakang. lalu dengan *raw image* pada *canvas UI* kita dapat menampilkan gambar yang kamera tersebut perlihatkan. Pada awalnya kita membuat *render texture* baru yang memiliki ukuran sama dengan *raw image*-nya. Pembuatan *texture render* baru dapat dilihat pada Gambar 4.12. Lalu dibuatlah kamera baru yang diletakan menghadap belakang dan diletakkan di tengah atas pada dalam mobil. Lalu kita mengganti value *target texture* menjadi *texture render* yang tadi kita buat. Pembuatan kamera dapat dilihat pada Gambar 4.13. Lalu dibuatlah *raw image* baru yang diletakkan sama dengan kamera yang tadi dibuat. Value *texture* pada *raw image* akan diganti dengan *render*

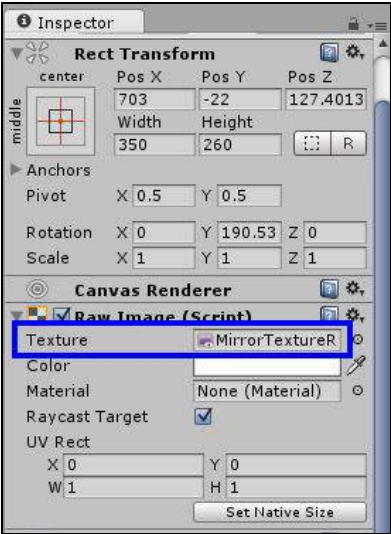
texture yang tadi telah dibuat. Pembuatan *raw image* dapat dilihat pada Gambar 4.14.



Gambar 4.12 Pembuatan *Texture Render Kaca Spion Kanan*



Gambar 4.13 Pembuatan Kamera Kaca Spion Kanan



Gambar 4.14 Pembuatan *Raw Image* Kaca Spion Kanan

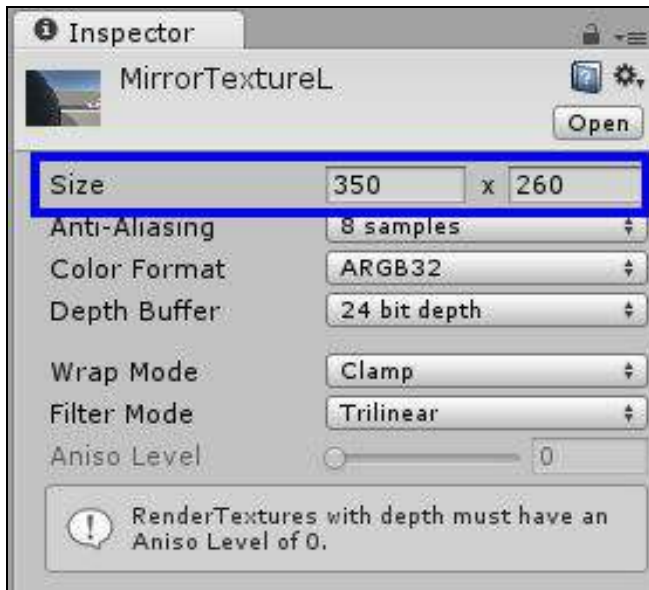
4.2.2.5 Implementasi Kaca Spion Kiri

Kaca spion kiri bertujuan untuk merepresentasikan kaca spion kiri yang berada pada mobil. Dengan adanya kaca spion kiri ini pengguna akan lebih mudah melihat bagian belakang kiri mobil. Tampilan kaca spion kiri dapat dilihat pada Gambar 4.15.



Gambar 4.15 Tampilan Kaca Spion Kiri

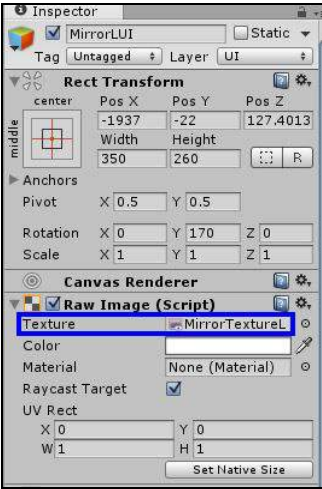
Kaca spion kiri ini di implementasikan dengan meletakkan kamera yang menghadap ke belakang sehingga kamera itu memperlihatkan seolah olah kaca yang menghadap kebelakang. lalu dengan *raw image* pada *canvas UI* kita dapat menampilkan gambar yang kamera tersebut perlihatkan. Pada awalnya kita membuat *render texture* baru yang memiliki ukuran sama dengan *raw image*-nya. Pembuatan *texture render* baru dapat dilihat pada Gambar 4.16. Lalu dibuatlah kamera baru yang diletakkan menghadap belakang dan diletakkan di tengah atas pada dalam mobil. Lalu kita mengganti value *target texture* menjadi *texture render* yang tadi kita buat. Pembuatan kamera dapat dilihat pada Gambar 4.17. Lalu dibuatlah *raw image* baru yang diletakkan sama dengan kamera yang tadi dibuat. Value *texture* pada *raw image* akan diganti dengan *render texture* yang tadi telah dibuat. Pembuatan *raw image* dapat dilihat pada Gambar 4.18.



Gambar 4.16 Pembuatan *Terxure Render* Kaca Spion Kiri



Gambar 4.17 Pembuatan Kamera Kaca Spion Kiri



Gambar 4.18 Pembuatan *Raw Image* Kaca Spion Kiri

4.2.2.6 Implementasi Layar Kamera Belakang

Layar Kamera Belakang bertujuan untuk merepresentasikan Layar yang menampilkan kamera belakang yang berada pada mobil. Dengan adanya layar belakang mobil ini pengguna akan lebih mudah melihat bagian belakang bawah mobil. Selain itu Pada layar ini juga menampilkan gigi saat ini pada mobil Tampilan layar belakang mobil dapat dilihat pada Gambar 4.19.



Gambar 4.19 Tampilan Layar Kamera Belakang

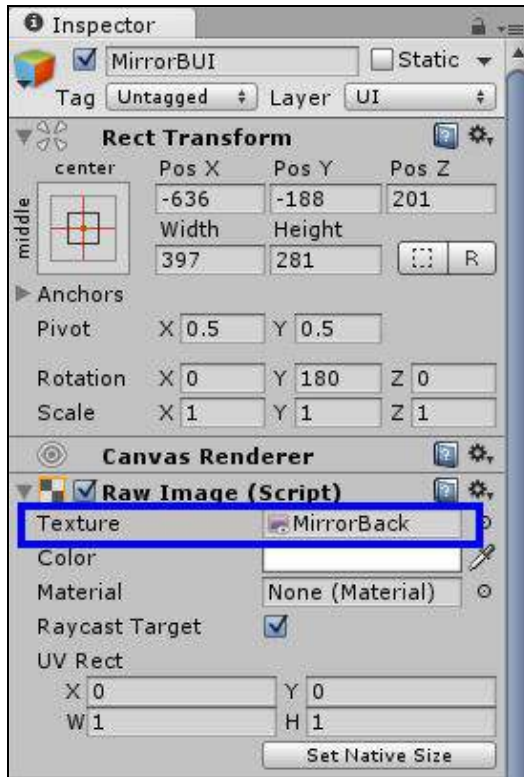
Layar belakang mobil ini di implementasikan dengan meletakkan kamera yang menghadap ke belakang pada belakang bawah mobil sehingga kamera itu memperlihatkan seolah olah menampilkan kamera yang menghadap kebelakang. lalu dengan *raw image* pada *canvas UI* kita dapat menampilkan gambar yang kamera tersebut perlihatkan. Pada awalnya kita membuat *render texture* baru yang memiliki ukuran sama dengan *raw image*-nya. Pembuatan *texture render* baru dapat dilihat pada Gambar 4.20. Lalu dibuatlah kamera baru yang diletakan menghadap belakang dan diletakkan di tengah atas pada dalam mobil. Lalu kita mengganti value *target texture* menjadi *texture render* yang tadi kita buat. Pembuatan kamera dapat dilihat pada Gambar 4.21. Lalu dibuatlah *raw image* baru yang diletakkan sama dengan kamera yang tadi dibuat. Value *texture* pada *raw image* akan diganti dengan *render texture* yang tadi telah dibuat. Pembuatan *raw image* dapat dilihat pada Gambar 4.22.



Gambar 4.20 Pembuatan *Texture Render* Layar Kamera Belakang



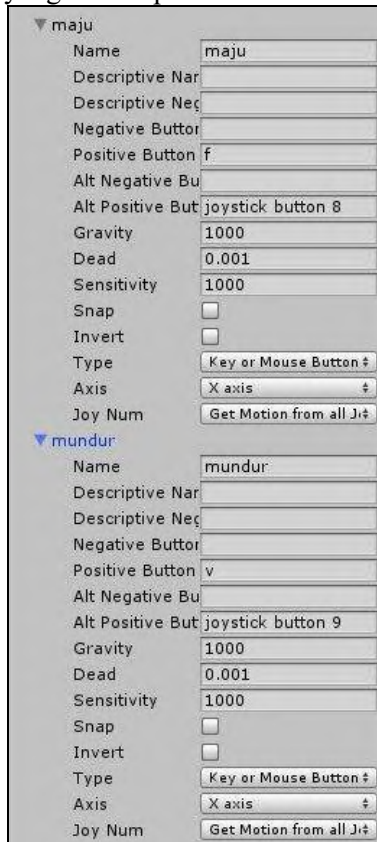
Gambar 4.21 Pembuatan Kamera Layar Kamera Belakang



Gambar 4.22 Pembuatan *Raw Image* Layar Kamera Belakang

Terdapat pula sebuah teks yang menunjukkan *gear* yang sedang dipakai saat itu juga. Terdapat 2 macam kondisi *gear* yaitu D dan R. Pada saat *gear* dalam kondisi D maka mobil hanya bisa maju. Sebaliknya pada saat *gear* dalam kondisi R maka mobil hanya bisa mundur. Untuk mengimplementasikan perubahan *gear* agar dapat digunakan di alat *steering wheel* ini diperlukan *input* baru yang dibuat di *input manager*. Tampilan *input* baru pada *input manager* dapat dilihat pada Gambar 4.23. Terdapat sumber kode yang diletakkan pada teks agar dapat berubah sesuai keinginan pengguna.

Implementasi perubahan teks *gear* pada mobil dapat dijelaskan pada Kode Sumber 4.3 yang berada pada kelas CarControl.



Gambar 4.23 Penambah *Input* Untuk Perubahan *Gear*

```
void UpdateGear()
{
    bool maju = Input.GetButton("maju");
    bool mundur = Input.GetButton("mundur");
    if (maju)
    {
        Tran = true;
    }
}
```



```

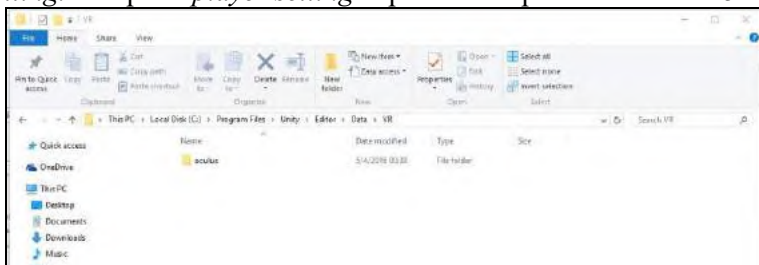
        GearShift.text = "Gear Shift : D";
    }
    else if (mundur)
    {
        Tran = false;
        GearShift.text = "Gear Shift : R";
    }
}

```

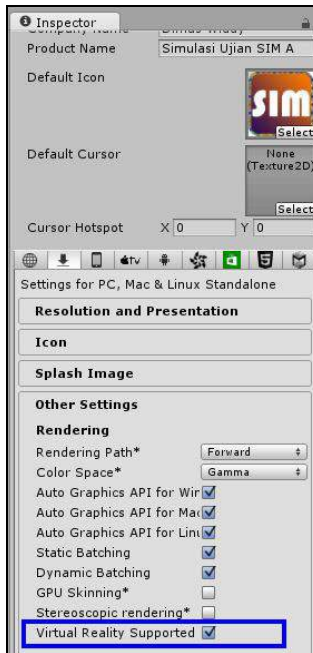
Kode Sumber 4.3 Kode Sumber Perubahan Teks *Gear*

4.3 Implementasi Integrasi Oculus Dengan Unity

Karena aplikasi ini menggunakan unity sebagai *platform* dan oculus rift sebagai alatnya, maka diperlukan integrasi agar oculus rift dapat digunakan saat membuat aplikasi ini. Implementasi integrasi oculus dengan unity pada saat membuat aplikasi ini adalah menggunakan OVRplugin for unity 1.3.2 yang dapat diunduh di [https://developer.oculus.com/downloads/\[3\]](https://developer.oculus.com/downloads/[3]). Hasil unduh dari berkas di atas adalah sebuah *package* yang berisi folder dan file. Terdapat 2 *folder* utama yaitu 5.3 dan 5.4. Karena pada implementasi aplikasi ini menggunakan unity 5.3.4f1 maka yang digunakan adalah *folder* 5.3. Dalam *folder* tersebut terdapat *folder* oculus yang akan dimasukkan ke direktori unity. Folder oculus tersebut lalu di *copy paste* ke direktori Unity/Editor/Data/VR. Tampilan direktori yang dipakai dapat dilihat pada Gambar 4.24. Setelah itu oculus siap digunakan pada unity. Hal terakhir yang dilakukan adalah mengizinkan unity untuk mengakses *virtual reality* pada *player setting*. Tampilan *player setting* dapat di lihat pada Gambar 4.25.



Gambar 4.24 Tampilan Direktori yang Digunakan

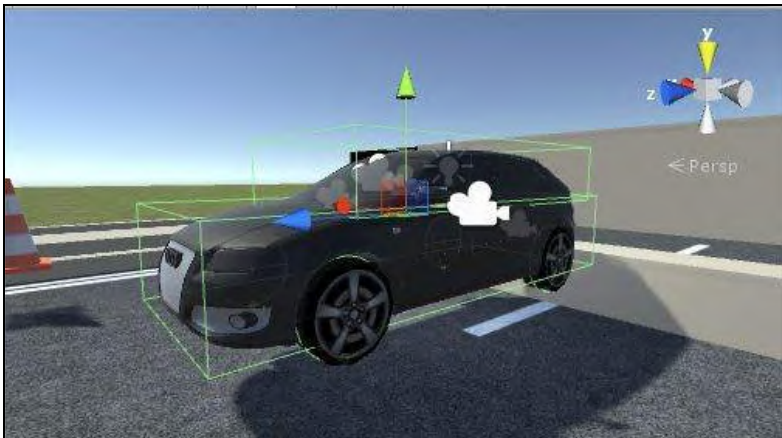


Gambar 4.25 Tampilan *Player Setting*

4.4 Implementasi Sistem Kerja Mobil

Karena aplikasi ini merupakan simulasi ujian SIM mobil, maka sistem kerja mobil sangatlah penting. Bagaimana mobil dalam aplikasi ini dapat berjalan sesuai dengan mobil sesungguhnya. Karena mobil adalah benda yang nyata maka diperlukan *collider* agar mobil yang dibuat dapat merasakan sentuhan terhadap objek yang lainnya. Terdapat 2 macam *collider* yang di implementasikan pada mobil, yaitu *box* dan *wheel collider*. *Box Collider* bertujuan untuk memberikan *collider* pada bagian atas atau bagan mobil. Lalu *wheel collider* berfungsi sebagai panyangga roda serta penggerak mobil. Sebagaimana roda menggerakkan mobil saat berjalan, maka *wheel collider* juga berfungsi sama. *Wheel collider* [8] memiliki fungsi yang dapat memberikan gaya baik bergerak maupun berhenti. Kedua gaya tersebut adalah *motorTorque* dan *brakeTorque*.

motorTorque adalah gaya yang di pakai untuk *wheelcollider* bisa bergerak dan brakeTorque adalah untuk berhenti. Setelah di coba dengan beberapa nilai, akhirnya diputuskan untuk menggunakan nilai 600 untuk motorTorque dan 60.000 untuk brakeTorque. Tampilan *Box* dan *wheel collider* dapat dilihat pada Gambar 4.26.



Gambar 4.26 Tampilan *Box* dan *Wheel Collider*

Tentu hal terpenting dalam mobil adalah bagaimana mobil itu dapat bergerak dan dikemudikan oleh pengguna sesuai dengan mobil yang sesungguhnya. Maka diperlukan kode agar mobil dapat bergerak sesuai dengan mobil sesungguhnya. Implementasi sistem kerja mobil dapat dijelaskan pada Kode Sumber 4.3 yang berada pada kelas CarControl.

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class CarControl : MonoBehaviour
{
    float MotorF = 600;
    float BrakeF = 60000;
```

```

        public WheelCollider[] wheelColliders = new
WheelCollider[4];
        public Transform[] tireMeshes = new Transform[4];
        public Transform centerOfMassBody;
        public GameObject SteerWheel;
        public Rigidbody car;
        bool Tran=true;
        public Text GearShift;
        public float accelerate;
        bool onetime = false;

        void Start()
        {
            GearShift.text = "Gear Shift : D";
            car.centerOfMass =
centerOfMassBody.transform.localPosition;

        }

        void Update ()
        {
            UpdateWheelRotation();
            UpdateGear();
            UpdateSteerAngle();
        }

        void FixedUpdate()
        {
            accelerate = Input.GetAxis("Vertical") +
Input.GetAxis("Vertical2");
            print(wheelColliders[0].motorTorque);
            UpdateWheelAngle();

            if (Tran == true)
            {
                wheelColliders[0].brakeTorque = 0;
                wheelColliders[1].brakeTorque = 0;
                wheelColliders[2].brakeTorque = 0;
                wheelColliders[3].brakeTorque = 0;
            }
        }
    }
}

```

```

        if (accelerate >= 0)
        {
            ForwardAccelerate();
        }
        else if (accelerate < 0)
        {
            Brake();
        }
    }

    else if (Tran == false)
    {
        BackwardAccelerate();
        if (accelerate >= 0)
        {
            wheelColliders[0].brakeTorque = 0;
            wheelColliders[1].brakeTorque = 0;
            wheelColliders[2].brakeTorque = 0;
            wheelColliders[3].brakeTorque = 0;
        }
        else if (accelerate < 0)
        {
            Brake();
        }
    }
}

void UpdateWheelRotation()
{
    for(int i = 0; i < 4; i++)
    {
        Quaternion quat;
        Vector3 pos;
        wheelColliders[i].GetWorldPose(out pos, out
quat);
        tireMeshes[i].position = pos;
        tireMeshes[i].rotation = quat;    }
    }
}

```

```

void UpdateGear()
{
    bool maju = Input.GetButton("maju");
    bool mundur = Input.GetButton("mundur");
    if (maju)
    {
        Tran = true;
        GearShift.text = "Gear Shift : D";
    }
    else if (mundur)
    {
        Tran = false;
        GearShift.text = "Gear Shift : R";
    }
}

void UpdateSteerAngle()
{
    float steer = Input.GetAxis("Horizontal") * -
90;
    float posy = car.transform.eulerAngles.y;
    float posx = car.transform.eulerAngles.x;
    SteerWheel.transform.eulerAngles = new
Vector3(posx, posy, steer);
}

void UpdateWheelAngle()
{
    float steer = Input.GetAxis("Horizontal");
    float finalAngle = steer * 50f;
    wheelColliders[0].steerAngle = finalAngle;
    wheelColliders[1].steerAngle = finalAngle;
}

void ForwardAccelerate()
{
    for (int i = 0; i < 4; i++)
    {

```

```

        wheelColliders[i].motorTorque =
accelerate * MotorF;
    }
}
void Brake()
{
    for (int i = 0; i < 4; i++)
    {
        wheelColliders[i].brakeTorque = -accelerate
* BrakeF;
    }
}
void BackwardAccelerate()
{
    for (int i = 0; i < 4; i++)
    {
        wheelColliders[i].motorTorque = -accelerate
* MotorF;
    }
}
}

```

Kode Sumber 4.4 Kode Sumber Sistem Kerja Mobil

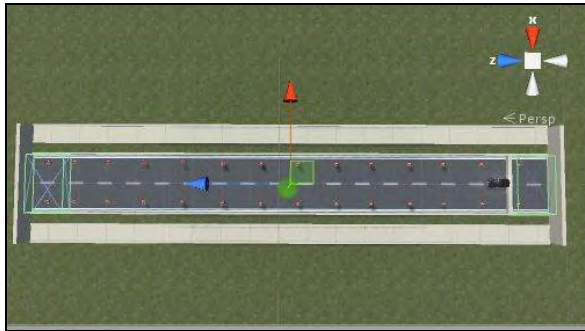
4.5 Implementasi Skenario Simulasi Ujian SIM

Dengan skenario simulasi ujian SIM, pengguna dapat mengetahui apa saja yang diujikan dalam ujian SIM sesungguhnya. Aplikasi ini mencoba untuk menyajikan 5 ujian yang ada pada ujian SIM sesungguhnya. Ujian tersebut antara lain adalah ujian maju mundur, ujian zig – zag, ujian parkir seri, ujian parkir parallel, dan ujian menaiki tanjakan.

4.5.1 Implementasi Skenario Ujian Maju Mundur

Implementasi pada skenario ujian maju mundur ini sama dengan perancangan pada Gambar 3.7. Maka terdapat kerucut jalan di kanan kiri dan sepanjang jalan. Lalu pada setiap ujung jalan. Pada awalnya teks navigasi bertuliskan “Majulah terus hingga ujung jalan.

Berhentilah sebelum dan jangan menabrak tembok”. Lalu di ujung jalan, terdapat *collider* yang berfungsi untuk merubah teks navigasi menjadi “Lalu mundurlah sampai ke ujung jalan yang lain”. Sehingga pengguna harus mundur ke ujung jalan yang lain. Pada ujung jalan belakang terdapat *collider* lain yang berfungsi untuk mendeteksi agar skenario dapat berhasil. Skenario dikatakan gagal apabila pengguna menabrakkan mobil ke kerucut jalan, tembok, maupun trotoar. Tampilan implementasi skenario ujian maju mundur dapat dilihat pada Gambar 4.27. Kode yang di pakai pada skenario ini dapat dilihat pada Kode Sumber 4.5



Gambar 4.27 Implementasi Skenario Ujian Maju Mundur

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class Ujian1 : MonoBehaviour {
    public WheelCollider wheelColliders;
    public Text navigasi;
    public GameObject area1;
    public GameObject area2;
    public GameObject area3;
    public Image background;
    public Text TextBerhasil;
    public Text TextGagal;
}
```



```

        public GameObject penghalangbelakang;
        public Rigidbody car;
        public Rigidbody car2;

        void OnTriggerStay(Collider other)
        {
            if (other.gameObject.name == "Area1" &&
car.velocity.magnitude < 0.01)
            {
                navigasi.text = "Lalu mundurlah
sampai ke ujung jalan yang lain";
                Destroy (penghalangbelakang);
                area3.transform.localScale = new
Vector3 (20, 20, 12);
            }
            else if (other.gameObject.name == "Area2" &&
car.velocity.magnitude < 0.01)
            {
                background.transform.localScale =
new Vector3 (1, 1, 1);
                TextBerhasil.transform.localScale
= new Vector3 (1, 1, 1);
                Time.timeScale = 0;
            }
        }
        void OnTriggerEnter(Collider other)
        {
            if (other.gameObject.tag == "Gagal")
            {
                background.transform.localScale =
new Vector3 (1, 1, 1);
                TextGagal.transform.localScale =
new Vector3 (1, 1, 1);
                Time.timeScale = 0;
            }
        }
        void OnCollisionEnter(Collision other)
        {
            if (other.gameObject.tag == "Gagal")
            {

```

```

        background.transform.localScale = new
Vector3(1, 1, 1);
        TextGagal.transform.localScale = new
Vector3(1, 1, 1);
        Time.timeScale = 0;
    }
    if (other.gameObject.tag == "Cone")
    {
        background.transform.localScale = new
Vector3(1, 1, 1);
        TextGagal.transform.localScale = new
Vector3(1, 1, 1);
        Time.timeScale = 0;
    }
}
}

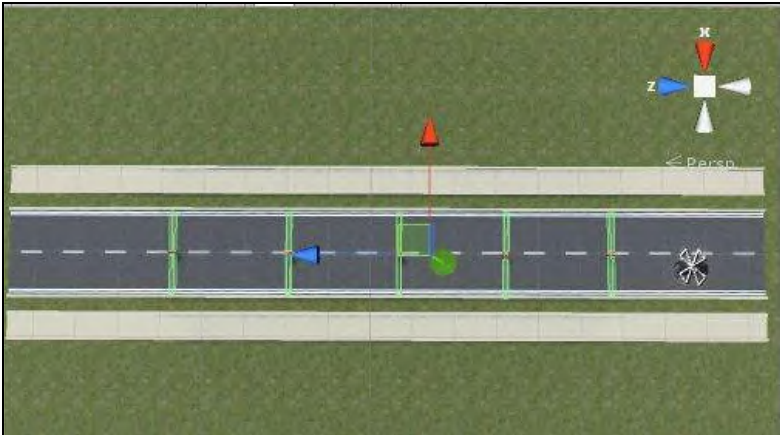
```

Kode Sumber 4.5 Kode Sumber Ujian Maju Mundur

4.5.2 Implementasi Skenario Ujian *Zig – Zag*

Implementasi skenario ujian *zig – zag* sama dengan perencanaanya yang dapat dilihat pada Gambar 3.8. Terdapat kerucut jalan yang terpasang lurus di depan mobil dengan memiliki jeda yang cukup besar tiap kerucutnya. Tujuan dari skenario ini adalah pengguna dapat mengemudikan mobil dengan melewati kerucut jalan secara bergantian. Pada awalnya teks navigasi bertuliskan “Lewatilah rintangan di depan dengan melewatinya dari sebelah kanan”. Terdapat 2 *collider* pada setiap kerucut jalan yang terdapat pada kiri dan kanannya. Apabila text navigasi bertuliskan melewatinya dari sebelah kanan, maka *collider* kanan bertujuan untuk merubah teks navigasi agar berubah menjadi “Lewatilah rintangan di depan dengan melewatinya dari sebelah kiri”. Lalu *collider* yang kiri berfungsi untuk menggagalkan skenario karena tidak menuruti apa yang ditulis di teks navigasi. Hal ini berlanjut sampai kerucut jalan habis. Saat kerucut jalan habis maka teks navigasi bertuliskan “Putar balik, lewatilah rintangan berikutnya

dengan melewatinya dari sebelah kanan”. Pengguna harus memutar balik arah mobil dan menuruti perintah teks navigasi seperti awal skenario. Tampilan implementasi skenario ujian *zig – zag* dapat dilihat pada Gambar 4.28. Kode yang di pakai pada skenario ini dapat dilihat pada Kode Sumber 4.6.



Gambar 4.28 Implementasi Skenario Ujian Zig-Zag

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class Ujian2 : MonoBehaviour {

    public Text textBerhasil;
    public Text textGagal;
    public Text navigasi;
    public GameObject background;
    public GameObject gagal1;
    public GameObject berhasil1;
    public GameObject gagal6;
    public GameObject gagal7;
    public GameObject gagal8;
```

```

public GameObject gagal9;
public GameObject gagal10;
public GameObject berhasil6;
public GameObject berhasil7;
public GameObject berhasil8;
public GameObject berhasil9;
public GameObject berhasil10;
public WheelCollider wheelColliders;
public GameObject success;
public Rigidbody car;

void Start ()
{

}
void Update ()
{

}
void OnTriggerEnter(Collider other)
{
    if (other.gameObject.tag == "Gagal")
    {
        background.transform.localScale =
new Vector3 (1, 1, 1);
        textGagal.transform.localScale =
new Vector3 (1, 1, 1);
        Time.timeScale = 0;

    }
    if (other.gameObject.name ==
"Berhasil1")
    {
        navigasi.text = "Lewatilah
rintangan dengan melewatinya dari sebelah kiri";
    }
    if (other.gameObject.name ==
"Berhasil2")
    {
        navigasi.text = "Lewatilah

```

```

rintangan berikutnya dengan melewatnya dari sebelah
kanan";
    }
    if (other.gameObject.name ==
"Berhasil3")
    {
        navigasi.text = "Lewatilah
rintangan berikutnya dengan melewatnya dari sebelah
kiri";
    }
    if (other.gameObject.name ==
"Berhasil4")
    {
        navigasi.text = "Lewatilah
rintangan berikutnya dengan melewatnya dari sebelah
kanan";
    }
    if (other.gameObject.name ==
"Berhasil5")
    {
        navigasi.text = "Putar balik,
lewatilah rintangan berikutnya dengan melewatnya
dari sebelah kanan";
        Destroy (gagal1);
        Destroy (berhasil1);
    }
    if (other.gameObject.name ==
"Putarbalik")
    {
        gagal6.transform.localScale = new
Vector3 (42, 50, 5);
        gagal7.transform.localScale = new
Vector3 (42, 50, 5);
        gagal8.transform.localScale = new
Vector3 (42, 50, 5);
        gagal9.transform.localScale = new
Vector3 (42, 50, 5);
        gagal10.transform.localScale =
new Vector3 (42, 50, 5);
        berhasil6.transform.localScale =

```

```

new Vector3 (42, 50, 5);
    berhasil7.transform.localScale =
new Vector3 (42, 50, 5);
    berhasil8.transform.localScale =
new Vector3 (42, 50, 5);
    berhasil9.transform.localScale =
new Vector3 (42, 50, 5);
    berhasil10.transform.localScale =
new Vector3 (42, 50, 5);
    success.transform.localScale =
new Vector3 (18, 7, 36);
    }
    if (other.gameObject.name ==
"Berhasil6")
    {
        navigasi.text = "Lewatilah
rintangan berikutnya dengan melewatinya dari sebelah
kiri";
    }
    if (other.gameObject.name ==
"Berhasil7")
    {
        navigasi.text = "Lewatilah
rintangan berikutnya dengan melewatinya dari sebelah
kanan";
    }
    if (other.gameObject.name ==
"Berhasil8")
    {
        navigasi.text = "Lewatilah
rintangan berikutnya dengan melewatinya dari sebelah
kiri";
    }
    if (other.gameObject.name ==
"Berhasil9")
    {
        navigasi.text = "Lewatilah
rintangan berikutnya dengan melewatinya dari sebelah
kanan";
    }

```

```

        if (other.gameObject.name ==
"Berhasil10")
        {
            navigasi.text = "Sekarang
berhentikan mobil";
        }
        void OnTriggerStay(Collider other)
        {
            if (other.gameObject.name == "Success"
&& car.velocity.magnitude < 0.01)
            {
                background.transform.localScale =
new Vector3 (1, 1, 1);
                textBerhasil.transform.localScale
= new Vector3 (1, 1, 1);
                Time.timeScale = 0;
            }
        }
        void OnCollisionEnter(Collision other)
        {
            if (other.gameObject.tag== "Cone")
            {
                background.transform.localScale =
new Vector3 (1, 1, 1);
                textGagal.transform.localScale =
new Vector3 (1, 1, 1);
                Time.timeScale = 0;
            }
        }
    }
}

```

Kode Sumber 4.6 Kode Sumber Skenario Ujian Zig-Zag

4.5.3 Implementasi Skenario Ujian Parkir Seri

Pada implementasi skenario ujian parkir seri, terdapat tempat parkir kosong yang dihipit oleh 2 mobil. Lalu pengguna dituntut untuk bisa memarkirkan mobilnya secara seri diantara mobil berwarna hijau dan putih dengan menghadap ke depan. Pada awal skenario teks navigasi bertuliskan “Parkirlah mobil secara 'seri'

menghadap depan di antara mobil berwarna hijau dan putih”. Terdapat *collider* di belakang tempat parkir yang berfungsi untuk tolak ukur keberhasilan skenario. Apabila bagian belakang dari mobil terkena *collider* tersebut, maka skenario dikatakan berhasil. Skenario dikatakan gagal apabila mobil yang dikendarai pengguna menabrak mobil lain, tembok, maupun trotoar. Terdapat 2 kelas kode yang digunakan pada skenario ini. Yaitu adalah kelas ujian3 dan ujian3collider. Kelas ujian3 berfungsi untuk mengecek mobil dengan kondisi gagal, sedangkan kelas ujian3collider berfungsi untuk mengecek mobil dengan kondisi berhasil. Tampilan implementasi skenario ujian parkir seri dapat dilihat pada Gambar 4.29. Kode yang di pakai pada skenario ini dapat dilihat pada Kode Sumber 4.7.



Gambar 4.29 Implementasi Skenario Ujian Parkir Seri

```
using UnityEngine;
using System.Collections;

public class Ujian3 : MonoBehaviour {

    public GameObject background;
    public GameObject TextGagal;
```



```

    void Start ()
    {

    }

    void Update ()
    {

    }

    void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.tag == "Gagal")
        {
            background.transform.localScale = new
Vector3(1, 1, 1);
            TextGagal.transform.localScale = new
Vector3(1, 1, 1);
            Time.timeScale = 0;
        }
    }

    void OnCollisionEnter(Collision other)
    {
        if (other.gameObject.tag == "Gagal")
        {
            background.transform.localScale = new
Vector3(1, 1, 1);
            TextGagal.transform.localScale = new
Vector3(1, 1, 1);
            Time.timeScale = 0;
        }
    }
}

using UnityEngine;
using System.Collections;

public class Ujian3Collider : MonoBehaviour {

```

```

public GameObject background;
public GameObject TextBerhasil;
public Rigidbody car;

    void Start ()
    {

    }

    void Update ()
    {

    }
    void OnTriggerStay(Collider other)
    {
        if(other.gameObject.name == "ColliderParkir"
&& car.velocity.magnitude < 0.01)
        {
            background.transform.localScale = new
Vector3(1, 1, 1);
            TextBerhasil.transform.localScale = new
Vector3(1, 1, 1);
            Time.timeScale = 0;
        }
    }
}

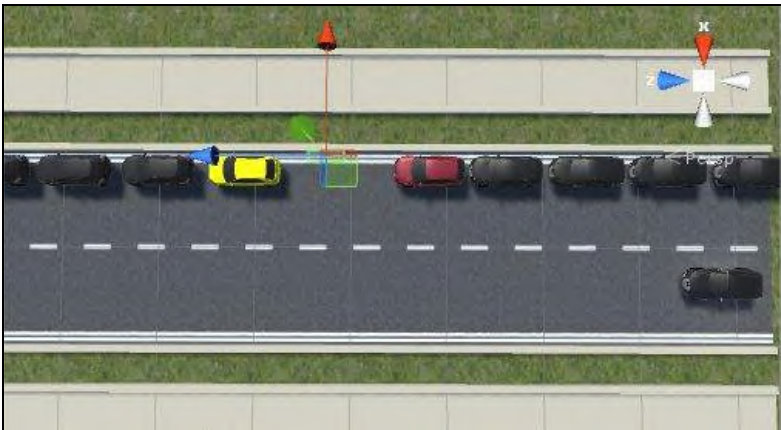
```

Kode Sumber 4.7 Kode Sumber Skenario Ujian Parkir Seri

4.5.4 Implementasi Skenario Ujian Parkir Paralel

Pada implementasi skenario ini, terdapat beberapa mobil yang berbaris dan parkir secara paralel. Pengguna disuruh untuk memarkirkan mobil diantara 2 mobil berwarna merah dan kuning yang parkir secara paralel. Pada awal skenario teks navigasi bertuliskan “Parkirlah secara 'parallel' diantara mobil merah dan kuning”. Sama seperti skenario ujian parkir seri, terdapat *collider* pada samping tempat parkir. Apabila mobil berhasil bersentuhan dengan *collider* tersebut maka skenario dikatakan berhasil. Skenario

dikatakan gagal apabila mobil yang dikendarai pengguna menabrak mobil lain maupun trotoar. Terdapat 2 kelas kode yang digunakan pada skenario ini. Yang pertama adalah kelas ujian4 dan yang kedua adalah ujian4collider. Kelas ujian4 berfungsi untuk mengecek mobil dengan kondisi gagal, sedangkan kelas ujian4collider berfungsi untuk mengecek mobil dengan kondisi berhasil. Tampilan implementasi skenario ujian parkir paralel dapat dilihat pada Gambar 4.30. Kode yang di pakai pada skenario ini dapat dilihat pada Kode Sumber 4.8.



Gambar 4.30 Implementasi Skenario Parkir Paralel

```
using UnityEngine;
using System.Collections;

public class Ujian4 : MonoBehaviour {

    public GameObject background;
    public GameObject TextGagal;

    void Start () {

    }

}
```

```

        void Update () {

        }
        void OnCollisionEnter (Collision other)
        {
            if(other.gameObject.tag == "Gagal")
            {
                background.transform.localScale = new
Vector3(1, 1, 1);
                TextGagal.transform.localScale = new
Vector3(1, 1, 1);
                Time.timeScale = 0;
            }
        }
    }

using UnityEngine;
using System.Collections;

public class Ujian4Collider : MonoBehaviour {

    public GameObject background;
    public GameObject TextBerhasil;
    public Rigidbody car;

    void Start()
    {

    }

    void Update()
    {

    }

    void OnTriggerStay(Collider other)
    {
        if (other.gameObject.name == "ColliderParkir"
&& car.velocity.magnitude < 0.01)
        {

```

```

        background.transform.localScale = new
Vector3(1, 1, 1);
        TextBerhasil.transform.localScale = new
Vector3(1, 1, 1);
        Time.timeScale = 0;
    }
}
}

```

Kode Sumber 4.8 Kode Sumber Skenario Ujian Parkir Paralel

4.5.5 Implementasi Skenario Menaiki Tanjakan

Pada skenario ini terdapat tanjakan tepat di depan mobil. Terdapat tembok dan teks stop pada tanjakan, sehingga pengguna harus berhenti disitu selama 5 detik lalu dapat melanjutkan menaiki tanjakan. Pada awal skenario, teks navigasi bertuliskan “Maju sampai tanjakan di depan dan berhentilah tepat di belakang tembok”. Terdapat *collider* tepat di belakang tembok sehingga apabila pengguna mengendarai mobil dan berhenti di belakang tembok, maka teks navigasi akan berubah menjadi “Tunggu Selama 5 detik”. Apabila mobil menabrak tembok maka skenario dikatakan gagal. Setelah menunggu 5 detik maka teks navigasi akan berubah menjadi “Naiki dan turunilah tanjakan di depan sampai di bawah, dan berhentilah di bawah”. Setelah itu pengguna pun disuruh untuk menaiki dan menuruni tanjakan. Setelah menuruni tanjakan, pengguna diminta untuk memberhentikan mobil yang dikendarainya. Setelah menuruni tanjakan dan berhenti di bawah, maka skenario dapat dikatakan berhasil. Skenario gagal apabila mobil yang dikendarai oleh pengguna menabrak tembok, maupun trotoar. Tampilan implementasi skenario ujian parkir paralel dapat dilihat pada Gambar 4.31. Kode yang di pakai pada skenario ini dapat dilihat pada Kode Sumber 4.9.



Gambar 4.31 Impelementasi Skenario Ujian Menaiki Tanjakan

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class Ujian5 : MonoBehaviour {

    public GameObject penghalang;
    public GameObject background;
    public Text TextBerhasil;
    public Text TextGagal;
    public GameObject area1;
    public Text navigasi;
    public WheelCollider wheelColliders;
    public Rigidbody car;
    public GameObject area3;

    void Start ()
    {

    }

    void Update ()
```

```

{
    }

    IEnumerator PenghalangDestroyer() {
        yield return new WaitForSeconds(5);
        Destroy (penghalang);
        navigasi.text = "Naiki dan turunilah tanjakan
di depan sampai di bawah, dan berhentilah di bawah";
    }

    void OnTriggerEnter(Collider other)
    {
        if(other.gameObject.name == "Area2")
        {
            navigasi.text = "Rem !";
        }
    }

    void OnTriggerStay(Collider other)
    {
        if (other.gameObject.tag == "Gagal")
        {
            background.transform.localScale = new
Vector3(1, 1, 1);
            TextGagal.transform.localScale = new
Vector3(1, 1, 1);
            Time.timeScale = 0;
        }
        if (other.gameObject.name == "Area1" &&
car.velocity.magnitude <0.01)
        {
            navigasi.text = "Tunggu Selama 5 detik";

            StartCoroutine(PenghalangDestroyer());
            Destroy(area1);
            area3.transform.localScale = new
Vector3(20, 20, 4);
        }
    }
}

```

```

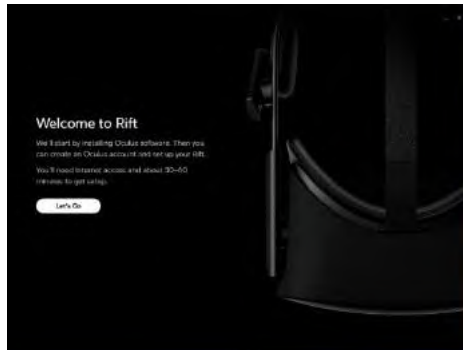
        if (other.gameObject.name == "Area2" &&
            car.velocity.magnitude < 0.01)
        {
            background.transform.localScale = new
Vector3(1, 1, 1);
            TextBerhasil.transform.localScale = new
Vector3(1, 1, 1);
            Time.timeScale = 0;
        }
    }
    void OnCollisionEnter(Collision other)
    {
        if (other.gameObject.tag == "Gagal")
        {
            background.transform.localScale = new
Vector3(1, 1, 1);
            TextGagal.transform.localScale = new
Vector3(1, 1, 1);
            Time.timeScale = 0;
        }
    }
}

```

Kode Sumber 4.9 Kode Sumber Skenario Ujian Menaiki Tanjakan

4.6 Petunjuk Instalasi

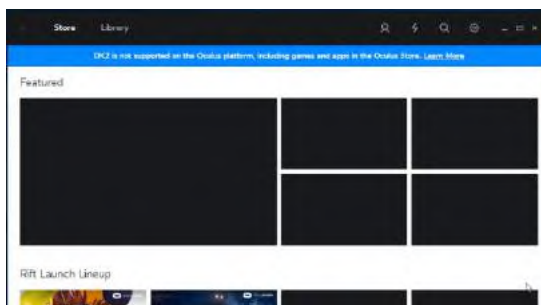
Pada subbab ini akan dijelaskan apa saja yang harus dilakukan agar aplikasi simulasi ini dapat dijalankan pada perangkat windows. Hal yang pertama adalah kita harus mempunyai instalasi Oculus Runtime sebagai *driver* sekaligus *launcher* utama dari oculus itu sendiri. Oculus Runtime ini sendiri dapat di download pada halaman download Oculus yaitu <https://developer.oculus.com/downloads/>. Setelah itu kita akan memulai proses instalasi Oculus Runtime. Tampilan pertama instalasi dapat dilihat pada Gambar 4.32.



Gambar 4.32 Tampilan Pertama Instalasi Oculus Runtime

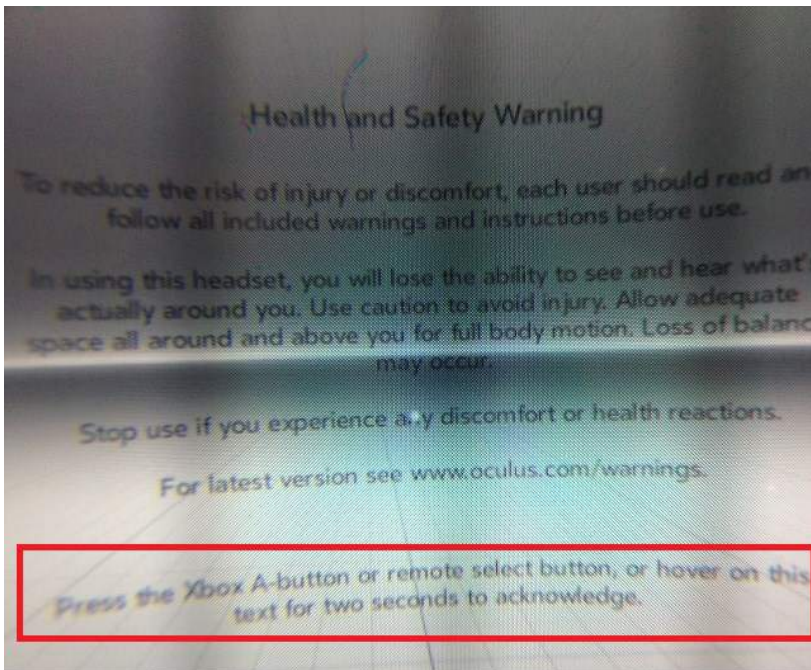
Setelah itu klik tombol 'Let's Go' dan akan muncul tampilan baru dimana pengguna disuruh untuk membaca *terms and condition* yang harus dipatuhi untuk memakai aplikasi ini. Bacalah ketentuan-ketentuan tersebut lalu klik tombol 'Agree' untuk melanjutkan. Lalu akan muncul tampilan *destination path*, klik 'Install Now' untuk melanjutkan. Setelah itu proses instalasi akan dimulai. Setelah instalasi selesai. Aplikasi simulasi siap untuk digunakan.

Untuk cara menjalankan aplikasi simulasi ini adalah dengan membuka oculus runtime terlebih dahulu. Apabila oculus runtime pertama kali dibuka, mak kita harus membuat akun oculus terlebih dahulu. Setelah membuat akun oculus, oculus runtime siap untuk dijalankan. Tampilan oculus runtime dapat dilihat pada Gambar 4.34.



Gambar 4.33 Tampilan Oculus Runtime

Kita hanya perlu menancapkan perangkat Oculus Rift dan *Steering Wheel* ke komputer. Oculus runtime harus tetap dinyalakan ketika kita ingin menjalankan aplikasi simulasi ini. Dalam keadaan menyala, nyalakan 'sim.exe' yang terdapat di folder 'Simulasi Ujian SIM A'. Lalu akan muncul peringatan kesehatan pada layar oculus rift. Kita hanya perlu meletakkan hover oculus rift ke bagian bawah, dan aplikasi dapat dijalankan. Tampilan peringatan kesehatan dapat dilihat pada gambar 4.33.



Gambar 4.34 Tampilan Peringatan Kesehatan Pada Oculus Rift

BAB V

PENGUJIAN DAN EVALUASI

Bab ini berisi bahasan mengenai uji coba dan evaluasi aplikasi simulasi ujian SIM dengan menggunakan *Steering Wheel* dan Oculus Rift. Uji coba dilakukan berdasarkan skenario yang telah ditentukan. Uji coba dilakukan terhadap hasil implementasi perangkat lunak yang dijelaskan pada Bab IV.

5.1 Lingkungan Uji Coba

Proses uji coba dilakukan pada lingkungan yang telah ditentukan. Pada uji coba ini, lingkungan dibedakan menjadi lingkungan perangkat keras dan lingkungan perangkat lunak. Tabel 5.1 menjelaskan mengenai tiap-tiap lingkungan uji coba aplikasi.

Tabel 5.1 Lingkungan Pengujian Perangkat Keras dan Lunak

Perangkat Keras	Prosesor : - Intel® Core™ i5-4460 Processor CPU@3.20GHz Memori : - 8 GB Kartu Grafis : - Nvidia GeForce GTX 960
Perangkat Lunak	Sistem Operasi : - Microsoft Windows 10 Pro 64-bit - DirectX 12 Perangkat Pengembang : - Unity 5.3.4f1 - Visual Studio Ultimate 2012

5.2 Skenario Pengujian

Pada skenario pengujian dijelaskan tentang skenario pengujian yang dilakukan. Metode pengujian akan mengacu pada *blackbox*

testing. Pengujian akan dibagi menjadi dua bagian, yaitu pengujian fungsionalitas dan pengujian aplikasi terhadap pengguna. Pengujian ini dilakukan untuk mengetahui apakah fungsionalitas sistem telah berjalan sebagai mana mestinya.

5.2.1 Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan dengan menyiapkan beberapa skenario pengujian sebagai tolok ukur keberhasilan pengujian dan mengacu pada kasus penggunaan yang sudah dijelaskan pada 3.1.2.1. Metode pengujian yang digunakan pada pengujian fungsionalitas mengacu pada *blackbox testing*.

5.2.1.1 Memilih Jenis Ujian SIM

Pengujian memilih jenis ujian SIM merupakan pengujian terhadap aplikasi untuk memilih jenis ujian sesuai dengan keinginan pengguna. Pengguna dapat memilih satu dari lima ujian yang disediakan. Skenario yang digunakan adalah pengguna disuruh untuk memilih salah satu dari lima ujian yang berada pada antarmuka halaman menu utama. Lima ujian tersebut adalah maju mundur, *zig-zag*, parkir seri, parkir paralel, dan menaiki tanjakan. Apabila pengguna memilih ujian pertama, maka seharusnya *scene* harus berpindah ke ujian pertama. Bergitu juga dengan ujian – ujian yang lainnya. Gambar 5.1 adalah contoh hasil pengujian memilih jenis ujian SIM. Tabel 5.2 akan menjelaskan lebih lanjut perihal pengujian memilih jenis ujian SIM.



Gambar 5.1 Kondisi Awal dan Akhir Pengujian Memilih Jenis Ujian SIM

Tabel 5.2 Pengujian Memilih Ujian SIM

Nomor	SP-UC-001
Referensi Kasus Penggunaan	UC-001
Nama	Memilih ujian SIM
Tujuan	Mengecek apakah aplikasi sudah dapat memilih ujian SIM sesuai dengan apa yang dipilih oleh pengguna saat antarmuka halaman utama
Kondisi Awal	Pengguna membuka aplikasi
Skenario	Pengguna memilih salah satu dari lima jenis ujian SIM yang ada
Keluaran yang Diharapkan	Aplikasi berpindah <i>scene</i> sesuai dengan ujian SIM yang dipilih oleh pengguna
Hasil Pengujian	Berhasil

5.2.1.2 Menjalankan Ujian SIM

Pengujian menjalankan ujian SIM merupakan pengujian terhadap aplikasi untuk dapat menjalankan skenario ujian SIM yang dipilih agar berfungsi sesuai dengan sebagaimana hasilnya. Terdapat enam skenario pada pengujian ini. Skenario yang pertama adalah pengguna disuruh untuk menyelesaikan skenario secara gagal. Skenario yang kedua sampai keenam adalah pengguna disuruh untuk menjalankan ujian sesuai dengan perintah sehingga ujian dikatakan berhasil. Kelima ujian yang di ujikan disini adalah ujian maju mundur, ujian *zig – zag*, ujian parkir seri, ujian parkir parallel dan ujian menaiki tanjakan Gambar 5.2 sampai 5.7 adalah contoh hasil pengujian memilih jenis ujian SIM. Tabel 5.3 sampai tabel 5.8 akan menjelaskan lebih lanjut perihal pengujian memilih jenis ujian SIM.



Gambar 5.2 Kondisi Awal dan Akhir Pengujian Menjalankan Ujian SIM Secara Gagal

Tabel 5.3 Tabel Pengujian Menjalankan Ujian SIM Secara Gagal

Nomor	SP-UC-002
Referensi Kasus Penggunaan	UC-002
Nama	Menjalankan Ujian SIM secara gagal
Tujuan	Mengecek apakah aplikasi sudah dapat memberikan hasil gagal apabila pengguna mengendarai mobil tidak sesuai dengan skenario. Contohnya adalah menabrak trotoar, menabrak kerucut jalan, menabrak mobil lain dan melewati jalan yang telah ditentukan.
Kondisi Awal	Pengguna memasuki <i>scene</i> skenario ujian
Skenario	Pengguna menabrak trotoar atau menabrak kerucut jalan atau menabrak mobil lain atau melewati jalan yang telah ditentukan.
Keluaran yang Diharapkan	Aplikasi berpindah <i>scene</i> yang menyatakan skenario ujian yang dilakukan telah gagal
Hasil Pengujian	Berhasil



Gambar 5.3 Kondisi Awal dan Akhir Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Maju Mundur

Tabel 5.4 Tabel Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Maju Mundur

Nomor	SP-UC-003
Referensi Kasus Penggunaan	UC-002
Nama	Menjalankan ujian SIM secara berhasil pada ujian maju mundur
Tujuan	Mengecek apakah aplikasi sudah dapat memberikan hasil berhasil apabila pengguna mengendarai mobil sesuai yang diperintahkan oleh teks navigasi pada ujian maju mundur.
Kondisi Awal	Pengguna memasuki <i>scene</i> skenario ujian maju mundur.
Skenario	Pengguna mengendarai mobil sesuai dengan teks navigasi. Pada pengujian ini adalah ujian maju mundur. Pengemudi akan mengendarai mobil maju sampai sebelum tembok, lalu mundur juga sampai sebelum tembok.
Keluaran yang Diharapkan	Aplikasi berpindah <i>scene</i> yang menyatakan skenario ujian yang dilakukan telah berhasil
Hasil Pengujian	Berhasil



Gambar 5.4 Kondisi Awal dan Akhir Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Zig-Zag

Tabel 5.5 Tabel Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Zig-Zag.

Nomor	SP-UC-004
Referensi Kasus Pengguna	UC-002
Nama	Menjalankan ujian SIM secara berhasil pada ujian <i>zig – zag</i>
Tujuan	Mengecek apakah aplikasi sudah dapat memberikan hasil berhasil apabila pengguna mengendarai mobil sesuai yang diperintahkan oleh teks navigasi pada ujian <i>zig – zag</i> .
Kondisi Awal	Pengguna memasuki <i>scene</i> skenario ujian <i>zig – zag</i> .
Skenario	Pengguna mengendarai mobil sesuai dengan teks navigasi. Pada pengujian ini adalah ujian <i>zig – zag</i> . Pengemudi akan mengendarai mobil melewati kerucut jalan yang ada di jalan secara bergantian. Pertama akan melewatinya dari kanan, maka setelahnya dari kiri. Begitu seterusnya hingga kerucut jalan berhasil dilewati semua.
Keluaran yang Diharapkan	Aplikasi berpindah <i>scene</i> yang menyatakan skenario ujian yang dilakukan telah berhasil
Hasil Pengujian	Berhasil



Gambar 5.5 Kondisi Awal dan Akhir Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Parkir Seri

Tabel 5.6 Tabel Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Parkir Seri

Nomor	SP-UC-005
Referensi Kasus Penggunaan	UC-002
Nama	Menjalankan ujian SIM secara berhasil pada ujian parkir seri
Tujuan	Mengecek apakah aplikasi sudah dapat memberikan hasil berhasil apabila pengguna mengendarai mobil sesuai yang diperintahkan oleh teks navigasi pada ujian parkir seri.
Kondisi Awal	Pengguna memasuki <i>scene</i> skenario ujian parkir seri
Skenario	Pengguna mengendarai mobil sesuai dengan teks navigasi. Pada pengujian ini adalah ujian parkir seri. Pengguna akan mengendarai mobil dan menempatkannya di antara mobil yang berwarna hijau dan putih secara paralel.
Keluaran yang Diharapkan	Aplikasi berpindah <i>scene</i> yang menyatakan skenario ujian yang dilakukan telah berhasil
Hasil Pengujian	Berhasil



Gambar 5.6 Kondisi Awal dan Akhir Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Parkir Paralel

Tabel 5.7 Tabel Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Parkir Paralel

Nomor	SP-UC-006
Referensi Kasus Penggunaan	UC-002
Nama	Menjalankan ujian SIM secara berhasil pada ujian parkir paralel
Tujuan	Mengecek apakah aplikasi sudah dapat memberikan hasil berhasil apabila pengguna mengendarai mobil sesuai yang diperintahkan oleh teks navigasi pada ujian parkir paralel
Kondisi Awal	Pengguna memasuki <i>scene</i> skenario ujian parkir paralel
Skenario	Pengguna mengendarai mobil sesuai dengan teks navigasi. Pada pengujian ini adalah ujian parkir paralel. Pengguna akan mengendarai mobil dan menempatkannya di antara mobil yang berwarna merah dan kuning secara paralel
Keluaran yang Diharapkan	Aplikasi berpindah <i>scene</i> yang menyatakan skenario ujian yang dilakukan telah berhasil
Hasil Pengujian	Berhasil



Gambar 5.7 Kondisi Awal dan Akhir Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Menaiki Tanjakan

Tabel 5.8 Tabel Pengujian Menjalankan Ujian SIM Secara Berhasil Pada Ujian Menaiki Tanjakan

Nomor	SP-UC-007
Referensi Kasus Penggunaan	UC-002
Nama	Menjalankan ujian SIM secara berhasil pada ujian menaiki tanjakan
Tujuan	Mengecek apakah aplikasi sudah dapat memberikan hasil berhasil apabila pengguna mengendarai mobil sesuai yang diperintahkan oleh teks navigasi pada ujian menaiki tanjakan
Kondisi Awal	Pengguna memasuki <i>scene</i> skenario ujian menaiki tanjakan
Skenario	Pengguna mengendarai mobil sesuai dengan teks navigasi. Pada pengujian ini adalah ujian menaiki tanjakan. Pengguna akan mengendarai mobil dan berhenti saat berada di tanjakan. Setelah menunggu 5 detik mobil pun maju dan menuruni tanjakan di depan.
Keluaran yang Diharapkan	Aplikasi berpindah <i>scene</i> yang menyatakan skenario ujian yang dilakukan telah berhasil
Hasil Pengujian	Berhasil

5.2.2 Pengujian Aplikasi Terhadap Pengguna

Aplikasi ini perlu dilakukan pengujian terhadap pengguna yang lainnya. Tujuannya adalah agar orang yang kurang tahu mengenai tes sim jadi tahu apa yang diujikan dalam ujian SIM A. Tujuan lainnya yaitu untuk mengetahui tingkat kemudahan dan kenyamanan aplikasi saat digunakan oleh pengguna. Metode pengujian yang dilakukan pada pengujian aplikasi terhadap pengguna mengacu pada *blackbox testing*.

Pengujian dilakukan kepada lima orang yang akan diminta untuk mengoperasikan aplikasi simulasi ujian SIM terlebih dahulu. Setelah itu pengguna diminta untuk mengisi kuesioner yang telah disediakan. Kuesioner dapat dilihat pada Tabel 5.9. Masing-masing kolom akan memiliki bobot penilaian tersendiri dengan ketentuan sebagai berikut :

- Sangat Setuju = 5
- Setuju = 4
- Cukup = 3
- Tidak Setuju = 2
- Sangat Tidak Setuju = 1

Tabel 5.9 Kuesioner Pengguna Simulasi Ujian SIM

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik					
2	Aplikasi memiliki menu yang					

	mudah digunakan					
3	Aplikasi memiliki tata letak tombol yang sesuai					
	Parameter Performa dan Kenyamanan					
4	Aplikasi nyaman untuk dimainkan					
	Parameter Materi Simulasi					
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya					
6	Saya rasa skenario pada aplikasi sama dengan sesungguhnya					

5.3 Evaluasi

Tahap evaluasi perlu di lakukan untuk mengolah hasil data yang telah dihasilkan. Lalu akan dihasilkan kesimpulan dan saran nantinya. Evaluasi dibagi menjadi dua bagian yaitu evaluasi pengujian fungsionalitas dan pengujian aplikasi terhadap pengguna.

5.3.1 Evaluasi Pengujian Fungsionalitas

Terdapat enam skenario pengujian fungsionalitas yang telah dilakukan pada tahap 5.2.1. Tabel 5.10 menunjukkan hasil dari pengujian fungsionalitas. Semua skenario pengujian telah dicoba dan berhasil. Jadi dapat disimpulkan bahwa fungsionalitas dari aplikasi ini dapat berjalan sesuai dengan harapan.

Tabel 5.10 Tabel Hasil Evaluasi Pengujian Fungsionalitas

ID	Nama	Hasil
SP-UC-001	Memilih Ujian SIM	Berhasil
SP-UC-002	Menjalankan Ujian SIM Pada Ujian Maju Mundur	Berhasil
SP-UC-003	Menjalankan Ujian SIM Pada Ujian Zig – Zag	Berhasil
SP-UC-004	Menjalankan Ujian SIM Pada Ujian Parkir Seri	Berhasil
SP-UC-005	Menjalankan Ujian SIM Pada Ujian Parkir Paralel	Berhasil
SP-UC-006	Menjalankan Ujian SIM Pada Ujian Menaiki Tanjakan	Berhasil

5.3.2 Evaluasi Pengujian Aplikasi Terhadap Pengguna

Aplikasi ini telah diujikan kepada lima orang pengguna. Setelah mereka mencoba aplikasi ini, mereka diberikan kuesioner yang di isi berdasarkan pendapat mereka. Rentan umur pengguna

adalah 20 sampai 22 tahun. Hasil dari Kuisioner dapat dilihat pada Tabel 5.11.


Tabel 5.11 Hasil Kuesioner Pengguna

No	Parameter Antarmuka	Nilai rata-rata
1	Aplikasi memiliki tampilan dan desain yang menarik	3.8
2	Aplikasi memiliki menu yang mudah digunakan	4.4
3	Aplikasi memiliki tata letak tombol yang sesuai	4
	Parameter Kenyamanan	
4	Aplikasi nyaman untuk dimainkan	3.6
	Parameter Materi Simulasi	
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya	3.8

6	Saya rasa skenario pada aplikasi sama dengan sesungguhnya	4.6
---	---	-----

Berdasarkan tabel diatas dibagi menjadi tiga bagian. Yaitu penilaian antarmuka, performa dan kenyamanan dan materi simulasi. Dimulai dari penilaian antarmuka, aplikasi ini mendapatkan rata – rata nilai 4.1. Nilai ini sudah lebih dari inilai 4 yaitu setuju. Hal ini menunjukkan bahwa pengguna setuju apabila aplikasi simulasi ujian SIM ini memiliki antarmuka yang menarik, mudah digunakan, dan sesuai. Lalu nilai kenyamanan mendapatkan nilai 3.6. Nilai ini sudah lebih dari inilai 3 yaitu cukup. Hal ini menunjukkan bahwa pengguna mengatakan bahwa aplikasi ini cukup nyaman digunakan, walaupun kurang setuju. Yang terakhir adalah nilai materi simulasi mendapatkan nilai 4.2. Nilai ini sudah lebih dari inilai 4 yaitu setuju. Hal ini menunjukkan aplikasi simulasi ujian SIM ini memiliki materi simulasi yang sesuai dengan sesungguhnya. Secara umum aplikasi ini mendapatkan nilai rata – rata 4 dari nilai maksimal 5.

LAMPIRAN


KUESIONER TUGAS AKHIR – 5112100144 DIMAS WIDDY PRATAMA

RANCANG BANGUN SIMULASI UJIAN SURAT IZIN MENGENUDI (SIM) MOBIL MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT

Identitas Responden

Nama Lengkap : Rahma Fida Usia : 21 Tahun

Pekerjaan : Nahasiswa Jenis Kelamin : ♀

A. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (√)

SS = Sangat Setuju S = Setuju C = Cukup

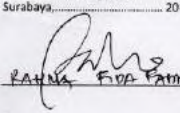
TS = Tidak Setuju STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi memiliki tata letak tombol yang sesuai			✓		
Parameter Performa dan Kenyamanan						
4	Aplikasi nyaman untuk dimainkan					✓
Parameter Materi Simulasi						
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya			✓		
6	Saya rasa skenario pada aplikasi sama dengan sesungguhnya					✓

B. KRITIK DAN SARAN

Tambahkan skenario seperti orang sedang menyeberang jalan

Surabaya, 10 Juni 2016


RAHMA FIDA PRATAMA

Gambar 0. 1 Kuesioner Responden Pertama



KUESIONER TUGAS AKHIR – S112100144 DIMAS WIDY PRATAMA

RANCANG BANGUN SIMULASI UJIAN SURAT IZIN MENGENUD (SIM) MOBIL MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT

Identitas Responden

Nama Lengkap
Pekerjaan

: Primo Sigil R
: Mahasiswa

Usia : 22 Tahun
Jenis Kelamin : P

A. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

C = Cukup

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan					✓
3	Aplikasi memiliki tata letak tombol yang sesuai					✓
Parameter Performa dan Kenyamanan						
4	Aplikasi nyaman untuk dimainkan			✓		
Parameter Materi Simulasi						
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya				✓	
6	Saya rasa skenario pada aplikasi sama dengan sesungguhnya					✓

B. KRITIK DAN SARAN


Tambahkan detail assets

Surabaya, 10 Juni 2016

Primo Sigil Rusnjaya

Gambar 0. 2 Kesioner Responden Kedua

97



ITS
Institut Teknologi Sepuluh Nopember

KUESIONER TUGAS AKHIR – 5112100144 DIMAS WIDDY PRATAMA

RANCANG BANGUN SIMULASI UJIAN SURAT IZIN MENGEMUDI (SIM) MOBIL MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT

Identitas Responden

Nama Lengkap : Angga Saputra D Usia : 22 Tahun
Pekerjaan : Mahasiswa Jenis Kelamin : L/P


A. PENILAIAN TERHADAP APLIKASI
Isilah tabel dibawah ini dengan menggunakan tanda (✓)
SS = Sangat Setuju S = Setuju C = Cukup
TS = Tidak Setuju STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan				✓	✓
3	Aplikasi memiliki tata letak tombol yang sesuai				✓	✓
Parameter Performa dan Kenyamanan						
4	Aplikasi nyaman untuk dimainkan			✓		
Parameter Materi Simulasi						
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya			✓		
6	Saya rasa skenario pada aplikasi sama dengan sesungguhnya			✓		✓

B. KRITIK DAN SARAN

← Steering wheel rotasinya terlalu pendek
Sedangkan mobil asli rotasinya melebihi 360° ke
kanan dan 360° ke kiri

Surabaya, 10 Juni 2016


ANGGA SAPUTRA

Gambar 0. 3 Kuesioner Responden Ketiga



KUESIONER TUGAS AKHIR – 5112100144 DIMAS WIDDY PRATAMA

RANCANG BANGUN SIMULASI UJIAN SURAT IZIN MENGENUDI (SIM) MOBIL MENGGUNAKAN STEERING WHEEL DAN OCULUS Rift

Identitas Responden

Nama Lengkap : Wahyu Widyamanda Usia : 21 Tahun
 Pekerjaan : Mahasiswa S1 Jenis Kelamin : L/P

A. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

C = Cukup

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi memiliki tata letak tombol yang sesuai				✓	
Parameter Performa dan Kenyamanan						
4	Aplikasi nyaman untuk dimainkan				✓	
Parameter Materi Simulasi						
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya					✓
6	Saya rasa skenario pada aplikasi sama dengan sesungguhnya				✓	

B. KRITIK DAN SARAN

Environmentnya diperbagus. Kalau bisa diadain 8th skenario
 juga karena jika dalam tes nyata tidak terpisah-pisah


Surabaya, 6 Oktober 2016

Wahyu

Wahyu Widyamanda

Gambar 0. 4 Kuesioner Responden Keempat

99



ITS
Institut Teknologi Sepuluh Nopember

KUESIONER TUGAS AKHIR – 5112100144 DIMAS WIDDY PRATAMA

RANCANG BANGUN SIMULASI UJIAN SURAT IZIN MENGENUDI (SIM) MOBIL MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT

Identitas Responden

Nama Lengkap : Fahmy T.H. Usia : 19 Tahun
Pekerjaan : Mahasiswa Jenis Kelamin : L / P


A. PENILAIAN TERHADAP APLIKASI
Isilah tabel dibawah ini dengan menggunakan tanda (v)
SS = Sangat Setuju S = Setuju C = Cukup
TS = Tidak Setuju STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik			✓		
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi memiliki tata letak tombol yang sesuai				✓	
Parameter Performa dan Kenyamanan						
4	Aplikasi nyaman untuk dimainkan			✓		
Parameter Materi Simulasi						
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya				✓	
6	Saya rasa skenario pada aplikasi sama dengan sesungguhnya				✓	

B. KRITIK DAN SARAN

Ditanya oleh SD sehingga semakin sesuai nyata karena banyak objek seperti di dunia nyata

Surabaya, 10 Juni 2016



Gambar 0. 5 Kuesioner Responden Kelima

BAB VI

KESIMPULAN DAN SARAN

Bab ini menjelaskan tentang kesimpulan yang didapatkan dari hasil pengujian yang telah dilakukan sebagai jawaban dari rumusan masalah yang telah dikemukakan sebelumnya. Setelah dari kesimpulan, diberikan juga saran sebagai pengembangan aplikasi kedepannya.

6.1. Kesimpulan

Dalam proses pengerjaan Tugas Akhir mulai dari tahap analisis, rancangan, implementasi, pengujian hingga evaluasi didapatkan kesimpulan sebagai berikut:

1. Ditinjau dari hasil kuesioner penguji, antarmuka aplikasi simulasi ujian SIM mendapatkan nilai 4.1 dari nilai maksimal 5 yang menunjukkan antarmuka yang menarik dan mudah dipakai. Lalu mendapatkan nilai kenyamanan 3.6 dari nilai maksimal 5 yang berarti aplikasi cukup nyaman dipakai, walaupun kurang nyaman. Dan mendapatkan nilai materi 4.2 dari nilai maksimal 5 yang menunjukkan pengguna setuju bahwa aplikasi memiliki materi ujian SIM sesuai sesungguhnya
2. Menurut ujian SIM yang sesungguhnya, terdapat lima buah ujian praktek. Yang pada aplikasi ini telah diterapkan skenarionya. Kelima ujian tersebut adalah ujian maju mundur, ujian *zig-zag*, ujian parkir seri, ujian parkir parallel, dan ujian menaiki tanjakan
3. Dengan memanfaatkan *collider* pada Unity, skenario ujian SIM dapat diimplementasikan dengan menggunakan *step* perintah yang divisualisasikan oleh teks navigasi. Dengan menuruti perintah dari teks navigasi, *collider* ditempatkan di tempat tujuan dan merubah perintah selanjutnya hingga pengguna menyelesaikan skenario ujian berdasarkan ujian SIM yang sesungguhnya.

4. Dengan menggunakan OVRplugin for Unity, kita dapat mengintegrasikan Oculus Rift dengan Unity. Dengan menyalin data yang berada pada *package* OVRplugin ke direktori Unity. Oculus Rift sudah siap dipakai untuk membuat aplikasi.

6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang, berdasarkan pada hasil perancangan, implementasi dan uji coba yang telah dilakukan.

1. Jenis *steering wheel* yang dipakai pada pengujian tidak mirip seperti *steering wheel* pada mobil sesungguhnya. Steering wheel yang digunakan hanya bisa memutar *steer* sebesar 90°. Mungkin dapat digunakan jenis *steering wheel* yang lain yang dapat digerakkan 360°.
2. Digunakan *assets* lingkungan yang lebih mirip gedung kantor polantas hingga mirip dengan ujian sesungguhnya.

DAFTAR PUSTAKA

- [1] X. Tang, Virtual Reality - Human Computer Interaction, InTech, Chapters published , 2012.
- [2] “Unity 3D - Game Engine | Herman Class,” [Online]. Available: <http://www.hermantolle.com/class/docs/unity-3d-game-engine/>. [Diakses May 2016].
- [3] TweakTown, “Genius Speed Wheel 6 MT Vibration Feedback Racing Wheel Review,” [Online]. Available: <http://www.tweaktown.com/reviews/5708/genius-speed-wheel-6-mt-vibration-feedback-racing-wheel-review/index.html>. [Diakses June 2016].
- [4] H. Setiawan. [Online]. Available: <http://setialheri.com/2015/06/10/sabar-menant-4-bulan-bersabar-akhirnya-lulus-juga-sim-a-di-satlantas-polres-gresik-jalur-reguler/>. [Diakses July 2016].
- [5] Oculus, “Oculus Rift Development Kit 2 (DK2) | Oculus,” [Online]. Available: <https://www3.oculus.com/en-us/dk2/>. [Diakses July 2016].
- [6] O. VR, “Developer Center - Downloads,” [Online]. Available: <https://developer.oculus.com/downloads/>. [Diakses May 2016].
- [7] Unity. [Online]. Available: <http://docs.unity3d.com/Manual/class-InputManager.html>. [Diakses May 2016].
- [8] Unity, “Unity - Scripting API: Wheel Collider,” [Online]. Available: <https://docs.unity3d.com/ScriptReference/WheelCollider.html>. [Diakses May 2016].

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis lahir di Surabaya, 24 September 1994, merupakan anak sulung dari dua bersaudara. Dalam perjalanan hidupnya penulis pernah menempuh pendidikan di SD Muhammadiyah GKB Gresik, SMP Muhammadiyah 12 GKB Gresik, SMA Negeri 1 Gresik dan S1 Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember (ITS) pada rumpun Interaksi Grafika dan Seni (IGS).

Selama menjadi mahasiswa penulis pernah bergelut di beberapa organisasi seperti HMTC ITS, Kepengurusan ITX Expo 2013, dan Schematics. Selain itu penulis juga pernah kerja praktik di PT Jabar Telematika Bandung. Penulis dapat dihubungi lewat widdydimas@gmail.com